NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

SAFETY OF MIXED MODEL ACCESS CONTROL IN A MULTILEVEL SYSTEM

by

Randall J. Arvay

June 2014

Dissertation Supervisor: James Bret Michael

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704–0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 2014

3. REPORT TYPE AND DATES COVERED
Dissertation

4. TITLE AND SUBTITLE: SAFETY OF MIXED MODEL ACCESS CONTROL IN A MULTILEVEL SYSTEM

5. FUNDING NUMBERS

6. AUTHOR(S) Randall J. Arvay

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943–5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. I.R.B. Protocol number___N/A___.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Information sharing can result in emergent behaviors that affect the safety properties associated with overt information flows. Secure cross-domain integration, involving the safety properties of both individual domains and the information dissemination across those domains, can result in leakage of information during the brokering of that information in an enterprise-level, multilevel secure (MLS) system using mixed model access control. Existing access control models do not address this problem. To address this gap, we developed a technique for building compositional models that combine both role-based access control and traditional MLS-based Bell-LaPadula models to provide for a high-assurance MLS system access controller. However, such compositional models introduce information rights proliferation during the specification of high-assurance security requirements and the security policy to provide for safety within the system. We addressed that problem with a technique that leverages RuleML to specify declassification policies for securing information exchange between different security levels of disparate access control models. The technique supports the tranquility principle allowing for desired information flows while not violating the overall security policy of the system. We demonstrated the technical feasibility of using both of these techniques, using as our example application cross-domain information sharing in conducting Maritime Domain Awareness operations.

14. SUBJECT TERMS RuleML, Cross-Domain, Multilevel Security, Maritime Domain Awareness, Information Leakage, Access Control, Information Broker

15. NUMBER OF PAGES
413

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

SAFETY OF MIXED MODEL ACCESS CONTROL IN A MULTILEVEL SYSTEM

Randall J. Arvay
Lieutenant Colonel, United States Army
B.S., United States Military Academy, 1993
M.S., Hawaii Pacific University, 2003

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN SOFTWARE ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2014

Author: Randall J. Arvay

Approved by:

James Bret Michael
Professor of Computer Science
Dissertation Supervisor

Dan C. Boger
Professor and Chair
Information Science

George W. Dinolt
Professor of Practice
Computer Science

Man-Tak Shing
Associate Professor of
Computer Science

Duminda Wijesekera
Professor of
Computer Science, George Mason University

Approved by: Peter J. Denning, Chair, Department of Computer Science

Approved by: Douglas Moses, Vice Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Information sharing can result in emergent behaviors that affect the safety properties associated with overt information flows. Secure cross-domain integration, involving the safety properties of both individual domains and the information dissemination across those domains, can result in leakage of information during the brokering of that information in an enterprise-level, multilevel secure (MLS) system using mixed model access control. Existing access control models do not address this problem. To address this gap, we developed a technique for building compositional models that combine both role-based access control and traditional MLS-based Bell-LaPadula models to provide for a high-assurance MLS system access controller. However, such compositional models introduce information rights proliferation during the specification of high-assurance security requirements and the security policy to provide for safety within the system. We addressed that problem with a technique that leverages RuleML to specify declassification policies for securing information exchange between different security levels of disparate access control models. The technique supports the tranquility principle allowing for desired information flows while not violating the overall security policy of the system. We demonstrated the technical feasibility of using both of these techniques, using as our example application cross-domain information sharing in conducting Maritime Domain Awareness operations.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF LISTINGS

LIST OF ACRONYMS AND ABBREVIATIONS

ABAC	attribute-based access control
ACL	access control list
ACM	access control matrix
AFRL	Air Force Research Laboratory
AIS	automated information system
ANOA	advance notice of arrival
APT	advanced persistent threat
BFM	boundary flow modeling
BIA	business intelligence and analytics
BLP	Bell & LaPadula model
BNF	Backus-Naur form
BPEL	business process execution language
C&A	certification and accreditation
CDS	cross-domain solution
CFG	context-free grammar
CONOPS	concept of operations
COTS	commercial-off-the-shelf
DAC	discretionary access control
DHS	Department of Homeland Security
DIB	defense industrial base
DMS	Defense Messaging System
DoD	Department of Defense
DSN	Defense Switched Network
DTG	date time group
ELINT	electronic intelligence
ESB	enterprise service bus
EWO	electronic warfare officer
FBI	Federal Bureau of Investigation
FW	firewall
GENSER	general service

GUI	graphical user interface
HAG	high-assurance guard
HGS	high-grade service
H-H-H	High-High-High assurance level
HRU	Harrison-Ruzzo-Ullman model
HTTP	hypertext transport protocol
HTTPS	hypertext transport protocol secure
HUMINT	human intelligence
IA	information assurance
IB	information broker
ICD	Intelligence Community Directive
ID	information declassifier
IDS	intrusion detection system
IP	Internet protocol
IPS	intrusion prevention system
ISSE	Information Support Server Environment
MAC	mandatory access control
MASINT	measure and signature intelligence
MDA	maritime domain awareness
MIEM	maritime information exchange model
MILS	multiple independent levels of security
MLS	multilevel secure systems
MMAC	mixed-model access control
MMSI	maritime mobile service identity
MYSEA	Monterey Security Architecture
NIEM	national information exchange model
NIPRnet	non-secure Internet protocol router network
NIST	National Institute of Standards and Technology
NRL	Navy Research Laboratory
NSA	National Security Agency
OMG	Object Modeling Group
PAP	policy administration point

PDP	policy decision point
PEP	policy enforcement point
PKI	public key infrastructure
RBAC	role-based access control
RBAC	rule-based access control
SCI	sensitive compartmented information
SIGINT	signals intelligence
SIPRnet	secure Internet protocol router network
SMTP	simple mail transport protocol
SOA	service-oriented architecture
TCB	trusted computing base
TDC	trusted database connector
TENCAP	tactical exploitation of national capabilities
TS	Top Secret
UDDI	universal description discovery and integration
UML	unified modeling language
URI	uniform resource identifier
VAR	vessel activity report
VOI	vessel of interest
WS	web services
WSBPEL	web services business process execution language
WSDL	web services description language
XACML	extensible access control markup language
XML	extensible markup language
XSD	XML schema definition

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Both the National Security Agency, through its Information Assurance Scholarship Program, and the United States Army have provided me with the tremendous opportunity to pursue doctoral studies at the Naval Postgraduate School. This degree and its process have only been made possible by the incredible support and guidance from many professors, several fellow students, and my family. To all of them, I am most appreciative.

I thank Professor Bret Michael for his unwavering support and commitment to me and to his many other students. He is a consummate professional in every aspect and I could not have asked for a better dissertation supervisor. My journey would not have been completed without the help of Professor Duminda Wijesekera. His invaluable guidance on RuleML and establishing proof of concept was a significant contribution to this research. He was always available to assist and was truly generous with his time and guidance. I particularly enjoyed our Friday afternoon sessions. He is truly a great educator and a phenomenal resource for any student. I would also like to thank the rest of my dissertation committee consisting of Professor Dan C. Boger, Professor George W. Dinolt, and Professor Man Tak Shing for their encouragement, support, guidance and perspective.

I thank Fred Glaser and Chris Newcomb for their support and funding to NPS for research efforts related to the Radiant Alloy program.

I thank my fellow Ph.D. students for their friendship and help in this process, particularly CDR Mike Schumann.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In his testimony to the Senate in February 2008, Vice Admiral John Michael McConnell, The Director of National Intelligence, offered:

> The U.S. information infrastructure including telecommunications and computer networks and systems, and the data that reside on them is critical to virtually every aspect of modern life. Therefore, threats to our IT infrastructure are an important focus of the Intelligence Community. As government, private sector, and personal activities continue to move to networked operations, as our digital systems add ever more capabilities, as wireless systems become even more ubiquitous, and as the design, manufacture, and service of information technology has moved overseas, our vulnerabilities will continue to grow. [1]

Protecting both the information infrastructure and the processing of data is a critical information assurance (IA) concern. Information sharing via an automated information system (AIS) requires that the AIS enforce confidentiality, integrity, and availability of the security policy[1] in the applicable domain.[2] Enforcing security policy is challenging when the data can flow between security domains and the information systems permit data at different levels of sensitivity to be accessed and stored on the same set of computing nodes. A cross-domain security architecture[3] delineates the allowable information flows between information systems. Consequently, such an architecture, if constructed, would support government and military information systems requiring multiple levels of security to support full spectrum operations in the ongoing Long War (formerly the Global War on Terror) and to meet the needs of our military and government organizations at the highest levels of IA. These multilevel secure systems

[1] A security policy is a set of rules that specify how information and resources are managed, protected, and distributed [2].

[2] A domain is a logical structure of resources or nodes working under the same security policy and management [2]. Examples of domains are: (1) a different classification level such as the Secret level (2) a separate management group such as the U.S. Army.

[3] Security Architecture is an architecture supporting the primary purpose of fulfilling security requirements in accordance with an established security policy to provide a predetermined level of trust [3].

require assurances[4] that they, in fact, offer protection of data and services between users, components and interfaces of varying levels of confidentiality, integrity and availability. Intelligence Community Directive (ICD) 503, "Information and Information System Governance" establishes the requirements and controls necessary for an information system to achieve hierarchically-defined levels of assurance, in addition to outlining the process for the certification and accreditation (C&A) of multilevel secure (MLS) systems [4]. The certification criteria for confidentiality, integrity, and availability to meet the updated ICD 503 High-High-High (H-H-H) assurance level have a stringent list of requirements for each facet. Developing systems to meet all of the security requirements of the policy directives, while also meeting all other types of requirements, is a challenge for software and systems engineers. There are two particular access control models in commercially available trusted products and DOD systems that will be used as the focus for this research: Bell-LaPadula (BLP) and role-based access control (RBAC). Information systems utilizing the BLP model are not sufficient to provide the level of data granularity and "need to share" capability required within a security domain. Nor is RBAC sufficient because it does not readily accommodate access control across multiple security domains. What is needed are MLS systems utilizing mixed-model access control, combining the features of the BLP and RBAC models to support net-centric enterprise services for sharing information [5].

One possible solution arose from a U.S. Navy Tactical Exploitation of National Capabilities (TENCAP) project named Radiant Alloy. In this type of mixed model access control system architecture, the Information Broker (IB) is an integral element. For an enterprise-level, service-oriented architecture (SOA)-based, MLS AIS-supporting mixed model access control, the IB plays the role of information management controller. The IB is the intermediary between the requester of the information and the data repositories. The IB provides the data and at the same time protects the anonymity of the source of the data. This anonymity is accomplished because the IB is effectively operating as a middle-man and collecting the data from multiple repositories. Thus, no attribution is linked to a

[4] Assurance is the confidence that an entity meets its security requirements, based on specific evidence provided by the application of assurance techniques [3].

particular source for any of the information response. The responsibility of the IB is "to facilitate the exchange of data between disparate applications" [6]. The IB is an architectural element that mediates access between differing data sources without requiring a custom connection, while also enabling the sharing of data between information systems, as depicted in Figure 1.

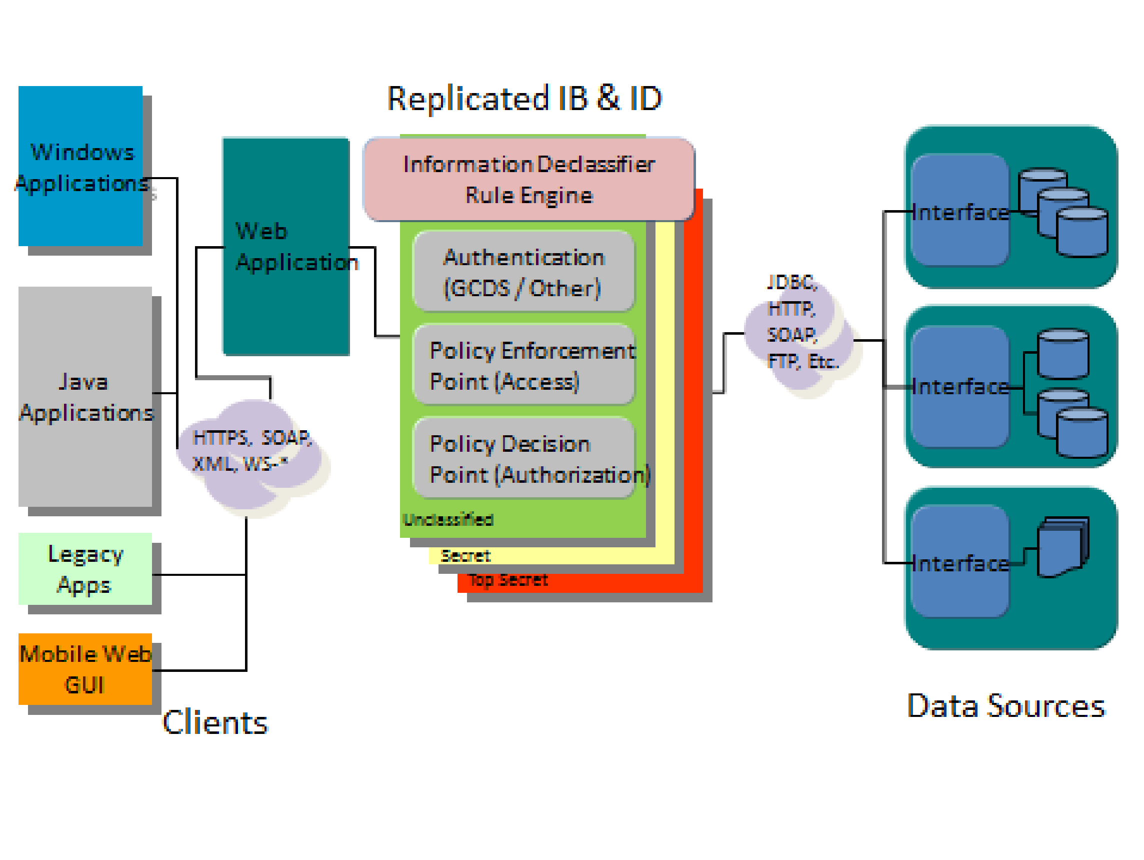


Figure 1. Information Broker System Architecture, after [7]

The IB orchestrates and logs all of the system-level access requests and responses. The IB mediates all user-level access requests and serves as a Policy Enforcement Point (PEP), if directed to do so by the Policy Decision Point (PDP). The PDP is responsible for authorization decisions, and resides in between the subjects, PEPs and the resource managers. The IB requests data through the Trusted Database Connector (TDC) and corresponding data stores to fulfill user requests. If required by policy, the IB must maintain the anonymity of the data origin, and orchestrating the ability of a user to write back to a data store that may be of a lower classification level (so the user cannot infer the origin). This involves the replication of the IB service at all sensitivity levels and having additional services that allow interaction between the IBs without alerting users and without impacting the inviolability of the data. The IB is also accessible from all possible platforms and environments, and provides mandatory access control for MLS based on a hierarchical, lattice-based access control model [6]. The information broker is intended to be a highly trustworthy component of a system, responsible for dealing with a myriad of clearances, classifications, and compartments.

Clearly, the IB is the central service of the system and must rely on the orchestration of several other services to fulfill its own role. However, each of these functions of the IB must be provided with some level of assurance that it meets the security requirements of ICD 503. The specification and management of access control is fundamental to the IB role. This role becomes more challenging to perform with the added complexity of having mixed access control models. The added complexity of relationships necessitates the establishment of guarantees against information leakage (non-interference) for a compositional[5] access controller. With this type of system, we discover a non-compositional property; that is, even if all components would not leak information individually, the combination thereof may enable information leakage via overt channels. This aspect of safety for mixed access controllers is an essential property for cross-domain MLS solutions. This research does not explore the leakage of information via covert channels, but rather provides for policy-based prevention of overt information flows that compromise the system’s information flows. The term “safety” as used here refers to protection provided by a system against leakage of information. Formal definitions of safety and other key terms are given in the background section of this chapter.

[5] Compositional refers to the combination of disparate access control systems and security policies into an aggregated information system.

A. STATEMENT OF THE PROBLEM

Information obtained first, or shared appropriately within an organization, helps an enterprise maintain a competitive advantage over adversaries, where real-time data feeds and data fusion aid in obtaining information. Other things being equal, information supremacy is essential to winning wars when the right information is shared at the right time, and the risk of sharing information with the wrong individuals is acceptably low to avoid catastrophic consequence [8]. With multiple information sources and the ever-increasing volume of data that is generated in real time (or near real time) and stored, enabling correlation and extracting information becomes a daunting challenge. The use of business intelligence and analytics (BIA) creates an opportunity for a real-time analytical engine to scan a broad set of unique data sources, identify common patterns in the data, translate those patterns into events, and then correlate and resolve those events to specific and actionable information. Traditional approaches to this big data analytic challenge have relied on batch retrieval and relational schemas combined into a structured analysis, but these do not account for the larger scale requirements of real-time streaming data and the correlation to find hidden relationships and useful information from seemingly useless data, as is found in a BIA tool.

The ability to gain superiority, either economically, politically, militarily, or otherwise through the processing and fusion of unclassified data and classified data to produce usable information that is shared across domains is the key impetus underlying the United States' National Security. For instance, Maritime Domain Awareness (MDA) is defined by the U.S. Department of Homeland Security as the

> effective understanding of anything associated with the global maritime domain that could impact the United States' security, safety, economy, or environment…MDA is a key component of an active, layered, maritime defense-in-depth. Maritime domain awareness will be achieved by improving our ability to collect, fuse, analyze, display, and disseminate actionable information and intelligence to operational commanders and decision makers. [9]

One arena where timely sharing of information is critical is cross-domain integration, which refers to subjects sharing information across two or more security domains. A real-world example of this integration lies with the distribution of intelligence indicators related to malware and Advanced Persistent Threat (APT) actors in the context of national security. This sharing between government, the defense industrial base (DIB), and other industry partners, like Symantec and McAfee, requires a cross-domain system to integrate the repositories of each proponent. Secure cross-domain integration requires safety of individual domains as well as safety of information dissemination across those domains. These domains may be governed by differing access control models and different security levels of the same MLS domain. Consequently, sharing information between them should be done so that it does not violate safety criteria of the access controller that governs the concerned domain and does not create covert channels enabling leakage of information. Information dissemination safety must work with disparate access controllers as well as varying declassification policies and rules between domains. Currently, we have many differing levels of security, but those levels are all separate. These systems are traditionally based on the BLP model and do not allow for information dissemination across domains. Human declassifiers and sneaker nets[6] can be used to transfer information between domains; however, these mechanisms are inefficient and impractical in today's highly connected environment hosting time-critical applications. Inherent with cross-domain information sharing is the risk of data leakage between domains. Each domain employs its own security policies and access control model, which can result in uncontrolled observability[7] between domains. This type of leakage violates information flow policy, as well as the safety of the system overall.

The current architectures using mixed model access controllers between domains are not sufficient and the singular examination of domains for safety is also not sufficient. The composition of access controllers can result in emergent violations of safety within

[6] Sneaker net refers to a manual work around for the transfer of information between information systems, where a user carries some type of physical media with information copied from one system to another for input.

[7] Observability refers to the visibility of the results of a change and in this research refers to how information in one domain can be inferred or viewed from outside of the domain or from separate permission subsets within the domain.

or between domains. It is from this lack of safety resulting from the composition of access controllers that we realize an information flow and information leakage problem that requires a re-architecting of the mixed model access controller to prevent the leakage. As a critical element of this re-architecting, we need to integrate an Information Declassifier (ID) element into the architecture to handle inconsistencies and emergent weaknesses in the security policy that arise from the composition of access control models. We must also develop and enforce specific rules and policies for the ID at every domain to allow for the desired level of information sharing while still protecting the safety of the system and the information flow. We propose using RuleML as a means to specify information flow control policies between security domains. Execution RuleML [10] specifies Prolog-like rules that use a XML-like syntax, and consequently are valuable for use in SOA. By re-architecting the mixed model access controller, and specifying the ID policies and rules, we can regain the safety of our information flows within the AIS and partially prevent information leakage between domains.

B. HYPOTHESIS

RuleML can be used as a policy specification language, based on a UML Use-Case analysis that supports mixed model access controllers for a high assurance, MLS, SOA-based system, to prevent information leakage and regain safety that is compromised by system behaviors that emerge from the composition of two or more different access control models.

This hypothesis will be tested with a proof of concept information broker and information declassifier ruleset developed to support the MDA Use-Case scenarios that use executable RuleML. The execution trace of the rules will demonstrate the efficacy of the IB to redirect information flows using RuleML and prevent information leakage.

C. BACKGROUND

Access control systems are expected to provide safety guarantees as the basis for trust and the overall assurances in the system. The access controls are responsible for managing all direct accesses to the system resources (objects[8]) according to the rules and assigned by the security policy and the access control model [11]. An access control model can be separated into many logical categories: discretionary access control (DAC), mandatory access control (MAC), Role-based Access Control Systems (RBAC), Attribute-Based Access Control Systems (ABAC), etc. These subject-centric access control models provide a framework that dictates how subjects can access[9] objects in the system. Other systems include capability-based systems, where each object has a list of capabilities that are required to be possessed by any user of the object. DAC systems allow the user to control access rights to the objects that they own, where this ability to change rights is subject to some set of rules, which can change during the course of operation of the system. Because of this, it is possible to bypass access restrictions and does not provide an assurance for the protection of data in the system. Once the user accesses data, the system can no longer control what the user does with the data (e.g., copy, move). The discretionary access control model then allows for the transfer or propagation of data in violation of the original security policy. Mandatory access control policies typically define a set of allowed access rights of subjects[10] to objects within a particular domain, and assume that the other objects are not allowed access. MAC policies are enforced by the system and not relegated to the user for access. However, there are systems specified using prohibitions, where the policy says which subjects are prohibited from accessing the resources, and all other subjects are allowed access. In MAC systems, subjects within the system are assigned sensitivity labels according to their level of trust, and objects are similarly assigned labels according to the level of trust that would be required of a subject to access the information in the object. A user is bound, during a specific session, to one or more subjects at a defined sensitivity level. This binding is constrained by the user's clearance level and the security level of the

[8] Objects are entities that contain or receive information.

[9] Access is defined as a subject's ability to perform some action such as read, write, copy, move, delete, and execute an object.

[10] Subjects are active entities, generally in the form of a person, program, process, or device that cause information to flow among objects, or that change the state of the system.

interface or terminal they are using, for the duration of that session. These subject labels are compared against that of an object, which the user might want to gain access to; if the subject label is equal to or higher than that of the object, the user may access the object. MAC models are representative of a MLS system, like a military security system based on the Bell-LaPadula model, where objects and subjects are ordered based upon their classification levels (Classified, Secret, etc.) and users are granted access to objects based on the relationship between the clearance level they possess, and the classification level of the object.

The safety guarantee relates to the aspect of information flow wherein information directly refused to a user cannot be indirectly obtained by executing a set of operations. Harrison et al. [12] defined authorization systems that allowed the modification of access rights, along with the ability for creating and deleting subjects and objects within the system. The safety concept introduced in [12] is that that access to an object within the system is impossible without the concurrence of the owner of that object. Since an owner in a DAC system may extend rights to an object that in turn may be given away without the owner's knowledge, no protection system can be safe by this definition. It is shown in [12] that it is generally undecidable whether "given an initial access matrix, there is some sequence of commands in which a particular generic right is entered at some place in the matrix where it did not exist before." Given an access matrix M and a right r (from a set of rights R), verifying the state of M with respect to r is undecidable. An additional aspect of safety that must be addressed for an MLS system is leakage from a covert channel. A covert channel utilizes shared resources in a system as a path of communication to transfer information. This is an unintended path from the original system design, but can be realized as either a storage channel or a timing channel [3]. In an information system, the non-existence of a covert channel cannot be proven. The amount of information that can be transmitted via a covert channel affects the severity of that channel to the security policy.

Bishop, in [3], defines secure versus safe with respect to the level of abstraction and implementation. Secure and non-secure are used to refer to the actual implementation of a system, while safety references the abstract security model[11]. Under these definitions, we can have a secure system that will correspond to an abstract model that is safe for all rights, but if we have a safe model, we do not necessarily ensure a secure system [3]. Bishop also adds to the definition of safety from [12] and further defines information leakage. Fundamental to these concepts is the access control matrix model, which is the simplest framework for describing a protection system. An access control matrix views a system in terms of the set of protected entities, contained in the set of objects O; subjects S is a set of active objects, such as users and processes; and the rights between subjects and objects drawn from a matrix A, where the set of rights R in each entry $a[s,o]$ where $s \in S$, $o \in O$, and $a[s,o] \subseteq R$. A matrix A, captures entity relationships where rights drawn from R get assigned to each entry $a[s,o]$. The protection states of a system are then represented by the triple (S, O, A). Leakage occurs when a generic right $r \in R$ is added to an element of the access control matrix not already containing r. The set of authorized states for the system are those in which no command or set of commands $c(x_1, \ldots, x_n)$ can leak r. A system is termed *safe with respect to the right r* if the system can never leak r [3]. Safety as described in [3] is critical for cross-domain solutions (CDS). CDS must employ access controls that guarantee safety in order to prevent the inadvertent transfer or disclosure of sensitive or classified information. Currently, no safety results exist for mixed model access controllers like that envisioned for use in the prototype system named Radiant Alloy. Through the mixed access control modeling of the security requirements[12] and the security policy[13] we can provide some assurances

[11] Security model is a framework that outlines the requirements necessary to properly support and implement a specific security policy.[2] The model depicts the logic and rules that must be implemented to support the security policy, and is a mapping of the abstract goals of the policy into rules that the system must follow.

[12] A security requirement consists of both functional requirements where it is a statement of some security function or security feature that should be implemented in a system; and a non-functional requirement which is a statement of a constraint or expected behavior that applies to a system, and may refer to the emergent properties of the software that is being developed or to the development process.[13]

[13] A security policy is a statement of what is, and what is not, allowed. [3]; a statement that outlines how entities access each other, what operations different entities can execute, what level of protection is required, and what actions should be taken when the requirements are not met.[2] The security policy outlines goals without regard for how they will be accomplished.

about the safety within the system. This will be done by a construction of safety by definition within the context of the system architecture using use, misuse, and security use cases.

A greater amount of work to verify the correctness of an MLS and a much higher cost is necessary due to the complexity of the system. A system is considered to operate in MLS mode when it permits two or more classification levels of information to be processed simultaneously and when all users do not have the appropriate clearance to access all of the information processed by the system [2]. An MLS system has added complexity in maintaining a separation of data and preventing unsafe or prohibited actions on objects within the system. The additional requirements necessary to meet a more stringent security policy and greater assurances required of a MLS system, affect the resulting security model and ultimately the implementation of that model in the final system. This complexity becomes apparent with the implementation of the access control model for the system itself. The greater complexity of the MLS access controller makes it harder to test and evaluate under all possible combinations of system accesses and subject to object pairings, and thus harder to provide assurance of its secure functionality. However, modeling methods have not kept pace with the rise in complexity, thus creating and exacerbating the separation of the system development and the underlying security of the system [14]. Additionally, those who are not security practitioners might view the added security requirements, necessary to meet a High assurance level, as an inconvenience that can be dealt with later. An enforcement and orchestration mechanism must be used to provide the integration of security policy and architecture in a SOA-based system. Access must be strictly controlled to enforce these security policies and maintain core data security. Assurance of the system’s functionality to support multiple levels of security hinges on four elements: the access control model, the security kernel, the information broker, and the information declassifier.

D. SIGNIFICANCE OF THE PROBLEM

MDA and similar national security related missions require sharing of information of multiple levels of sensitivity across security enclaves. This has become an evolving challenge to maintain necessary information flows as current access control models are not sufficient to support desired cross-domain solution (CDS) requirements. The compositional model introduces emergent challenges for information rights proliferation (i.e., information leakage) as we model high-assurance security requirements and the security policy to provide for safety within the system. In this research, we specify safety properties of the mixed model access controller and the rules necessary to implement an information declassifier using a UML-based Use-Case security analysis and use RuleML as a means to specify information flow control policies between security domains. Ruleset checks are also provided to demonstrate that the safety property for information flows still holds with the addition of the information declassifier to the system architecture. By following this approach, we can re-architect an access controller to ensure security requirements are achieved without the specified policy violations or leakage of information resultant from the compositional model, thus regaining safety.

E. RESEARCH APPROACH

The process flow planned for conducting this research is shown in Figure 2 and includes the following steps:

1. *Develop a Concept of Operations as a Basis for a Security Policy*–develop realistic Use Case and Misuse Cases that exercise the system requirements and challenge the security specifications for a mixed model access control system using the BLP and RBAC models.
2. *UML-based Use Case Analysis to Generate a Security Use Case and Re-architecting Development*–determine the Security Use Case necessary to prevent, mitigate, and detect the Misuse Cases while still permitting all Use Cases, to maintain the safety of information flows within our system.

3. *System Re-architecting Development*–determine the re-architecting necessary to meet the Security Use Case requirements.
    a. Provide Sequence Diagrams for the Security Use case in the revised architecture.
    b. Develop sample data sets as a basis for the underlying system model.
    c. Develop sample Vessels, Queries and Alerts using XML
    d. Define Information Declassifier rules using RuleML
    4. *Safety Refinement*–demonstrate what constitutes safety in our mixed model access control system and how the architecture supports the concept based on the re-architecting.
5. *Rule Verification, Regression Testing, and Proof of Concept Simulation*–use template system in a standard Maritime Domain Awareness scenario as a proof of concept and demonstrate technical feasibility of the verification of the re-architecting and the specification of the security policy using RuleML in a simulation of the various scenarios.

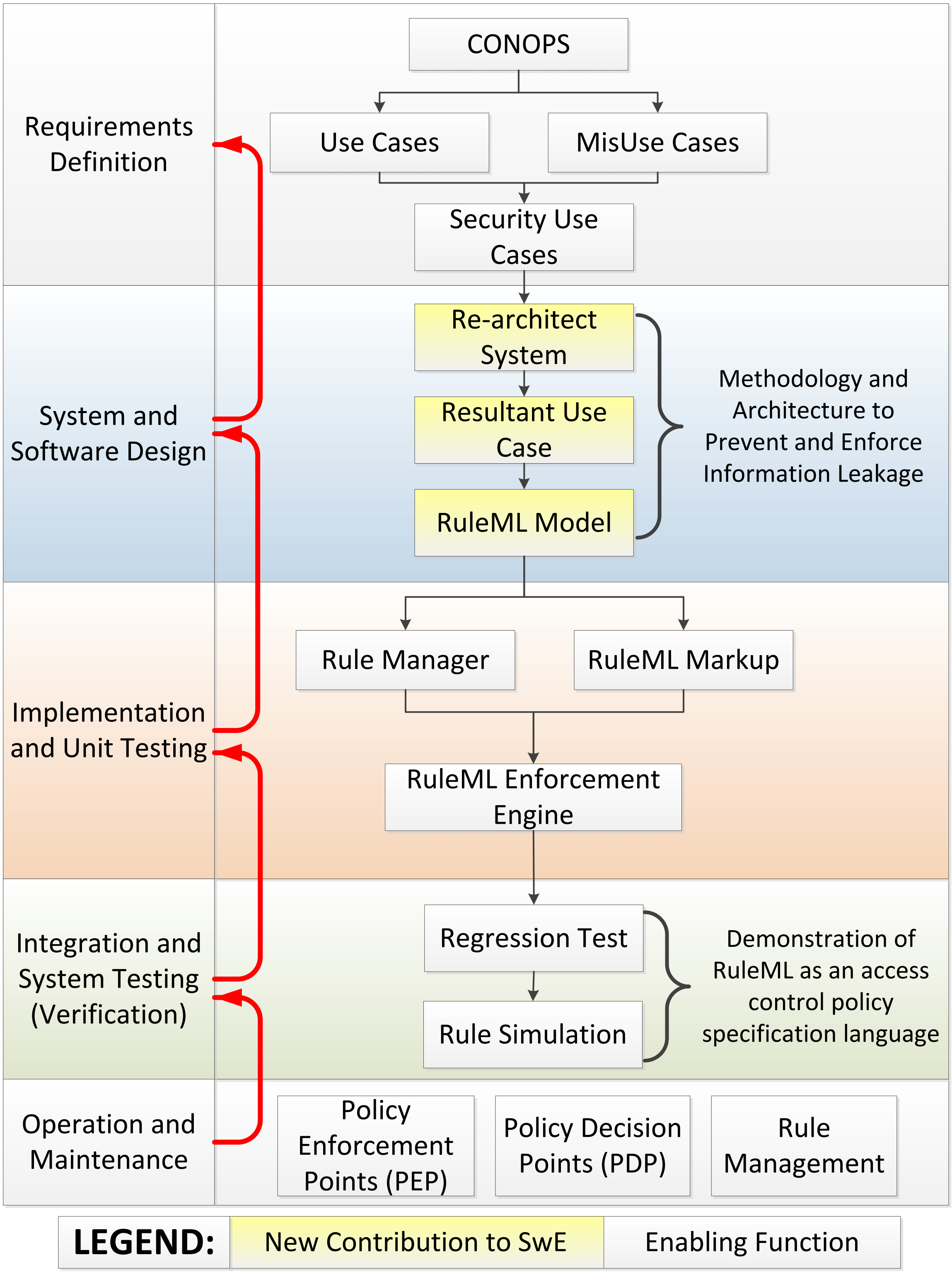

Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing (Verification)
Operation and Maintenance
CONOPS
Use Cases
MisUse Cases
Security Use Cases
Re-architect System
Resultant Use Case
RuleML Model
Methodology and Architecture to Prevent and Enforce Information Leakage
Rule Manager
RuleML Markup
RuleML Enforcement Engine
Regression Test
Rule Simulation
Demonstration of RuleML as an access control policy specification language
Policy Enforcement Points (PEP)
Policy Decision Points (PDP)
Rule Management
LEGEND:
New Contribution to SwE
Enabling Function

Figure 2. Methodology and Approach Flow

F. CONTRIBUTIONS OF THIS RESEARCH

As described above, the composition of access controllers in a system may result in emergent behaviors that violate the safety property of that system. We propose a methodology for modeling these behaviors and re-architecting a system to detect, or mitigate these emergent behaviors. We also provide a sample framework that validates the re-architecting and uses a rule-based service to prevent the overt information flows and regain safety within our system. We demonstrate the adequacy of RuleML for modeling and reasoning about access control security policy in a cross-domain context. The specific contributions of this work are listed below:

1. Developed and Used a New Methodology for Security Analysis using UML-based Use Case Analysis to Direct the Re-architecting of a System for the Preservation of Safety

The methodology of security analysis using UML Use and Misuse Cases, allows for tailoring of security policies and mitigating the “highest cost” risks for information flow while still enabling trusted sharing. UML-based Use Case security analysis has not been applied in a MLS, SOA-based venue and this work utilizes that approach as an exemplar to provide for a greater access to information and increased sharing among disparate security domains. In order to support inviolate information flows to the security policy, and to overcome the tranquility property associated with traditional MLS systems, we demonstrate that Radiant Alloy needs to use an information declassifier.

2. Developed a New Process to Allow for the Prioritization of Information Flows and the “Safe” Composition of Mixed Access Control Models through a Re-architecting of the System

Through the process of prioritizing the information-flow needs within a system, we can tailor the composition of different access control models to support required flows. This effectively opens the information sharing infrastructure to more DOD organizations and provides a means for cooperation and more real-time passing of information between agencies. This process also helps engineers and developers combine “best-of” practices in their design and implementation of an AIS. The idea of prioritization is based on the risk analysis and risk tolerances of the stakeholders. The allowable “exceptions” to our baseline policies are considered as the only permissible extensions and ultimately determine what is safe for the system. The method shown in this research uses RuleML to enforce this policy. The need for sharing and the need to maintain security must be balanced to provide an operationally useful system that will still uphold the desired and specified safety for information flows.

3. Created a New Process for the Development of Business Process Rules, using RuleML to Specify Allowed Information Flows and to Restrict Flows Enabling Leakage of Information from Emergent Behaviors and Facilitate Sharing between Information Systems

To support the development of an information declassifier, we provide a policy-based framework to do so using RuleML [15]. By developing, implementing and enforcing business process rules through the use of RuleML, we realize a greater control of our information flows. The rule engine provides for a greater granularity of Need (Not) to Share for information within domains, facilitating the dissemination of information and providing increased speed of information flow to allow a greater use of “real-time” information, compared with traditional MAC-only policies. Current systems cannot support this level of information sharing and traditionally use a workaround (i.e., sneakernet) to meet user requirements. The extension of an open-source XML-based business process language to facilitate the composition of access controllers and the safety of the associated models provides a useful tool for software engineers in system-architecting efforts.

4. Provided a Process that Allows Engineers to use Decision Tables or Other Methods to Develop Simple Propositions to Express Desired Process and Information Flows that can be Formed into Executable RuleML

With the re-architecting of an information system, it is necessary to provide a convincing argument that the RuleML specification supports the system’s required information flows. By creating business process rules with RuleML to support the sharing of information within the system, we can show that all prior capabilities and intended flows are still available to the user, yet there are no unintended flows that result in violation of the safety property. This is an extension of the work conducted on the hook-up theorem for multilevel security with respect to inference control and the composability of restrictiveness for security policies [41]. This must be done to build a basis of confidence for services to be used (and reused) with respect to the Information Broker and its associated rule engine based Information Declassifier service. By regression testing, which includes running the ruleset with a sample data set, we can verify that our security use case is correct. This is necessary to help establish a basis for certification of information systems at high assurance levels. The rule-based, Information Declassifier service helps to automatically provide Information Broker web services with a means of information sharing and information filtering that is not available within individual access control models, and to ensure the safety of information flows in mixed model access control systems. This included the validation of the concept system re-architecting through the conduct of a regression testing verification and a simulation of RuleML content-based query injection and filtering, whose results verified the rule structure and usage of RuleML as a specification language.

G. OVERVIEW OF THE DISSERTATION

In Chapter II, we provide background information on several areas essential to understanding this research, as well as cover an assessment of previous work in the respective areas.

Chapter III includes an introduction to the operational context that we used as a basis for the work. The service-oriented, cross-domain secure system is outlined as it pertains to this research.

Chapter IV introduces a motivating example of why information in a cross-domain context is necessary in Maritime Domain Awareness. The Use Case scenario is depicted in detail, which is the basis for the re-architecting and security analysis to preserve safety. We describe the desired and undesired flows and the rules necessary to allow and prevent those flows accordingly.

Chapter V highlights the queries system and alert messaging that are used in our prototype system. We describe the scenario-based query process and show examples with differing Web Services (WS) languages, along with the specification of Alert Messaging to support the scenario.

Chapter VI provides a summary of the re-architecting methodology and its subsequent rule-based execution from our prototype system. We briefly discuss the Security Use Case information flows that were required for the scenario and how the Information Declassifier supported those requirements. We also provide a prototype of the Information Declassifier re-architected system through a RuleML policy specification and its execution.

Chapter VII highlights the contributions toward software engineering and information assurance as well as future directions of our research.

H. KEY FINDINGS

In this research, we show how the methodology for UML-based Use Case Analysis can be used to elicit and model undesirable information flow properties within a mixed access control model, SOA-based, MLS system. This provides a systematic approach to regaining safety and enabling trusting sharing and desired information flows, without violating tranquility. We also show how this analysis can be used to re-architect a system to detect, mitigate, or prevent undesirable information flows, and extend this re-architecting by adding rules, via RuleML, to maintain and enforce the changes in system architecture as a security policy specification language. By using RuleML, we show the feasibility of this approach that can be extended to other technologies or languages to specify policy and information flows. Our work provides for composing disparate access control model systems, allowing for an accurate, security-grounded process for policy management. Composing different systems with diverse access control models can result in information leakage, and consequently, result in the composed system violating some safety criteria. We have shown how this can be addressed by introducing an information declassifier, of which the de-classification policies can be written in RuleML. The use of regression testing provides verification of the content-based query injection and filtering for the RuleML concept ruleset. The functionality of the Information Declassifier mandates a response during each step of the rule firing order, requiring the use of reactionary rules in RuleML. Both positive case and negative case rules were established and linked via forward chaining throughout the ruleset to create and enforce the policy model. By preventing an ambiguous or a non-response from within the ruleset, we can be assured that the Information Declassifier will also return a result. The MDA scenario provides a motivating example for cross-domain information sharing. Finally, we demonstrate the use of RuleML rules to maintain information flow safety within a system through simulation, which verified the rule structure and usage.

THIS PAGE INTENTIONALLY LEFT BLANK

II. ASSESSMENT OF PREVIOUS WORK

The research described in this dissertation builds on a variety of prior efforts to construct multilevel secure systems, enable information sharing, and realize the effective implementation of security policies with the support of a service-oriented architecture. This work also builds upon the research of McDaniel and Tardy [16] on role-based access control for MDA and that of Bennett [17] on defining a common intelligence picture for MDA. While a large amount of work has been done related to covert channels and the exploration of blocking those channels via different methods, the use of and mitigation of covert channels is not explored in this dissertation [2, 18-22]. This research covers overt information flows and the prevention of information leakage resultant from the composition of disparate systems.

A. ACCESS CONTROL

Access controls are used to verify that desired user accesses to system objects are authorized. Part of these access controls are done with an overarching policy or model, while the secondary piece, and arguably a highly important one as well, is the enforcement of the policy via an access control mechanism. The policy is used to specify allowable actions within a system and will be the focus of this research rather than the implementation or enforcement mechanism. The access control policy ensures that information does not flow from one set of subjects to another in the system (e.g., Top Secret information flowing to Unclassified), and that there is no path that exists between any two subjects through some combination of objects or even other subjects. Because of these desirable restrictions, access control must also discern between various users or subjects authorizations, and the objects (processes, files, data, domains, etc.) that they require access to. Also required is the ability to protect the subjects and objects themselves from improper use or modification. With many security models we find an underlying access matrix forming the basis for the access control policy.

Many access control matrices exhibit the idea of ownership for objects within the system. In our SOA-based system, we do not rely on or implement an ownership based access because of the Information Broker. The IB services user requests and acts as our access control mechanism. Inherent with this is the idea that the IB or the originating data store will have ownership of an object and not individual subjects (users). This eliminates the ability of subjects to grant or revoke privileges to objects, since they never have ownership.

1. An Access Control Matrix

The access control matrix (ACM) is a representation of policy and the system at a given state. The access control matrix model is typically regarding as the simplest framework for describing a protection system. An access control matrix views a system in terms of the set of protected entities, contained in the set of objects *O*; subjects *S* is a set of active objects, like users and processes; and the rights between subjects and objects drawn from a matrix *A*, where the set of rights *R* in each entry $a[s,o]$ where $s \in S$, $o \in O$, and $a[s,o] \subseteq R$. A matrix *A*, captures entity relationships where rights drawn from *R* get assigned to each entry $a[s,o]$. The protection states of a system are then represented by the triple (S, O, A). Objects typically include files, memory space, and even processes. Subjects are typically users, processes, or domains (i.e., privileged). Each of the rights specified within the matrix differentiates the types of actions that can be performed on the respective objects. The resultant mapping of *Subjects* to *Objects*, with *Rights* associated is an access control matrix. In [12], six primitive operations are defined:

Enter *r* into $A[s, o]$
Delete *r* from $A[s, o]$
Create subject *s*
Create object *o*
Destroy subject *s*
Destroy object *o*

These primitive operations were shown in [12] to be decidedly safe, since they are monotonic and that for non-monotonic systems safety is much more complex and they represent changes in state for the system in question. Monotonicity refers to the system in question only conducting one operation at a time and this is where the safety can be shown. It is these states of the system itself that we can relate to safety. By not allowing a system to reach an unauthorized state, we have effectively limited the model to safe behavior. However, a safe system is not necessarily a secure system. Safety refers to the model of the system, while security refers to the implementation of that model in the actual system. This is the fundamental difference and security is what is measured during the C&A process.

2. Access Control Matrix Safety

Any system that allows for information sharing will have information flow leakage. This leakage is often a permissible type, where users will trust other users with access rights to data. The safety guarantee that we need for our systems, relates to the aspect of information flow wherein information directly refused to a user cannot be indirectly obtained by executing a set of operations. Harrison et al. [12] define authorization systems allowing the modification of access rights, along with the ability for creating and deleting subjects and objects within the system. The safety concept introduced, stated that access to an object within the system was impossible without the concurrence of the owner of that object. Since an owner in a DAC system may extend rights to an object that in turn may be given away without his knowledge, no protection system can be safe by this definition. They subsequently showed that it is generally undecidable whether, “given an initial access matrix, there is some sequence of commands in which a particular generic right is entered at some place in the matrix where it did not exist before” [12]. Thus, given an access matrix *M* and a right *r* (from a set of rights *R*), verifying the state of *M* with respect to *r* is undecidable. An additional aspect of safety of concern to a MLS system is leakage from a covert channel. A covert channel utilizes shared resources in a system as a path of communication to transfer information. This is an unintended path from the original systems design, but can be realized as either a storage channel or a timing channel [3].

3. Safety Analysis

The safety of an access control matrix has been shown, in general, to be undecidable. Generally, the complexity (of the security verification) is tied to the choice of the security model’s abstraction level, and the design principles of the security architecture for the implementation.

The overarching security policy drives the choice of the security model. In this case, the need to provide access right proliferation across boundaries, leads to selection of the Harrison-Ruzzo-Ullman (HRU) model. HRU provides access control matrices combined with state machines to enable security property analysis through the observation of state transitions. For a *safety* analysis of this, we want to know if it is possible, given an initial matrix (*M*), that a *subject (s)* can obtain a *right (r)* to an *object (o)* in *M*. This would entail a leakage problem, wherein access (via the rights) to an object by a subject is gained, without the concurrence of the object's owner.

Harrison-Ruzzo-Ullman analysis has been shown to be decidable when the model is restricted (e.g., mono-operational operations). But, we cannot effectively model current security policies with such a restricted model. Mono-operational systems are easier to model, but, not as useful or as expressive as required to meet the modeling needs of a complex policy. Based on the reduction methods proposed, if we can in fact discount the entire matrix and only represent sub-matrices based on domain, then we could gain a reduction in state space. This reduction in complexity, if reduced sufficiently, could then be used with fully automated safety analysis.

Despite differing security domains, web services via SOA can be used to integrate these multiple domains, multiple policies, and multiple enforcement mechanisms into a usable framework. The RBAC model is a candidate for implementing this policy in a single domain, but a compositional model that extends RBAC and uniformly integrates between domains would be ideal.

B. MANDATORY ACCESS CONTROL MODELS

Mandatory access control models are the most stringent for access to objects and resources in a system.

1. Bell and LaPadula Model

The Bell-LaPadula (BLP) Model is associated with military-style classification of information and is used to enforce rules to provide confidentiality protection [3, 23]. The BLP model was developed to address the security of time-sharing mainframe systems and the leakage of sensitive information, and particularly to prevent sensitive information from being accessed in an unauthorized manner. The BLP model is a state machine model that enforces the confidentiality aspect of access control, where the state machine is the mathematical basis to show that the model (machine) will begin and remain in a secure state at all times. The BLP model defines the legal transitions. The Basic Security Theorem, is "if a system initializes in a secure state and all allowed state transitions are secure, then every subsequent state will be secure"[2]. This theorem is based on two fundamental conditions of the model.

The definition of the Simple Security condition, given in [23], states that a user can only access objects assigned an equal or lower classification level to that he or she is cleared for; this concept is commonly referred to as "read down." It can be expressed as: S (Subject) can read O (Object) if and only if $lo \le ls$ (l represents the security clearance) and S has discretionary read access to O [3].

The *-Property (Star Property) states that, to ensure that more sensitive information is not moved to less sensitive objects by malicious software, a subject cannot write into an object that is of a lower classification level; this rule permits "write up" but not "write down" [23]. It can be expressed as: S can write O if and only if $ls \le lo$ and S has discretionary access to O [3].

In the BLP model, all objects in the system are assigned a sensitivity level based on their classification level, and all users are similarly assigned a clearance level. Each clearance represents a sensitivity level and all are linearly ordered (e.g., Unclassified, Confidential, Secret, Top Secret). This model works well for generic access to an entire information domain (e.g., Secret), but to further segment usage rights we need to tailor the permission for objects even further within the domain, which allows us to extend the model by adding categories within each classification; this is generally done by using a lattice as shown in [3, 24] to support the "need to know" for users within a classification or through the use of role-based access control within each domain to generate our mixed model access control. Lattices can compose partial or total ordering among the elements of a set, where S is a finite set of elements and R is a relation. A simple linear ordering of security classes, shown in Figure 3, falls into a lattice structure, where S is a finite set of elements (e.g., S = {Unclassified, Secret, Top Secret}), and R is a relation (e.g., R = {Unclassified < Secret < Top Secret}).

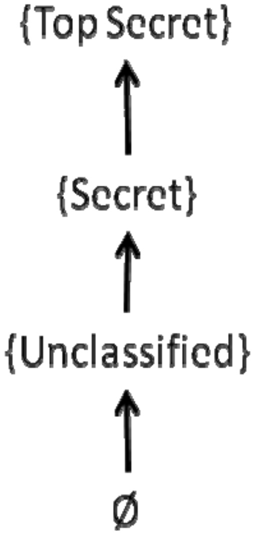


Figure 3. Linearly Ordered Lattice, from [3]

Lattices can compose partial or total ordering among the elements of a set, where S is a finite set of elements (e.g., S = {x, y, z}), and R is a relation of those elements. This nonlinear structure allows for greater complexity in the structure and for composing structures with ordered sets and subsets among non-related objects [24]. Denning’s nonlinear lattice example is shown in Figure 4 [3].

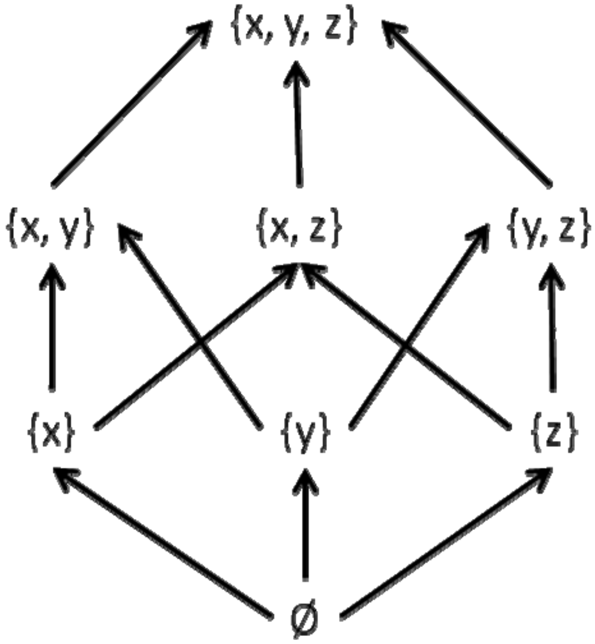


Figure 4. Lattice Demonstrating Non-Linear Ordering, from [3]

By adding this structure, the flow of information can be controlled and secure flows can be specified.

2. Role-Based Access Control

Role-based access control (RBAC) is another mandatory access control model that uses subjects, roles and permissions, as primitive entities and two mappings; subject-to-role mapping and role-to-permission mapping. The model maps a subject to the roles assigned to that subject and the role to the previously designated permission set. In this model, a subject obtains permission to act on an object based on permission assigned to the role that subject fills. Each role r is authorized to perform transactions in support of the role. This set of actions is defined as, *trans*(*r*). The active role that a subject is performing is defined as, *actr*(*s*), while the authorized roles for a subject is shown as, *authr*(*s*). A Boolean predicate detailing whether a given subject *s* can execute a transaction *t*, is shown by *canexec*(*s*, *t*) [3]. Given these basic definitions, we can construct axioms to detail the rest of the model. The rule of role assignment shows that if a subject can execute any transaction, then that subject has an active role;

$(\forall s \in S)(\forall t \in T)[canexec(s,t) \rightarrow actr(s) \neq \varnothing]$ [3], where S is set of all subjects and T is the set of transactions [3]. The rule of role authorization states that a subject must be authorized to assume its active role; $(\forall s \in S)[actr(s) \subseteq authr(s)]$ [3]. This rule prevents a subject from assuming any arbitrary role, and thus executing any transaction authorized to that arbitrary role. The rule of transaction authorization prevents a subject from executing a transaction which its current role is not authorized; $(\forall s \in S)(\forall t \in T)[canexec(s,t) \rightarrow t \in trans(actr(s))]$ [3]. Each of these three rules serves to restrict the transactions that can be performed within the RBAC system. Additional roles can be introduced to account for the domination of roles and for the concept of separation of duty for roles. The domination or containment rule can be expressed as; role r_i contains role r_j, where r_i>r_j and $(\forall s \in S)[r_i \in authr(s) \wedge r_j > r_i \rightarrow r_j \in authr(s)]$ [3] The use of a separation of duty rule is critical for a military AIS in which a single user cannot maintain all permissions. This rule can be expressed using two roles, r_1 and r_2 where $(\forall s \in S)[r_1 \in authr(s) \rightarrow r_2 \notin authr(s)]$ [3]. RBAC can be effectively used for limiting access where the subject's need to know must be further restricted within a security level. RBAC has also been shown in [25] to help provide safety guarantees for the leakage problem and for the separation of roles and subject authorizations to access data. RBAC provides for the enforcement of least privilege within a system. Least privilege is a design principle that guides the overall design of a system and impacts on the choice of security access policy used and its enforcement. Privileges are allocated to roles and then users are assigned to those roles. Interfaces may be constructed that are available to only certain subsets of the user population. An audit mechanism may provide separate interfaces for the audit manager, the audit operator, and the audit reviewer. The interfaces provide the least privilege that the user needs to complete his or her job, because each of the interfaces would provide different levels of functionality upon login and authentication.

In addition, least privilege can be used for the internal structure of the system. One aspect is to construct modules so that only the elements encapsulated by the module are directly operated upon. Elements external to a module that may be affected by the module’s operation are indirectly accessed through interaction (e.g., via a function call or a service request) with the module that contains those elements. Another aspect of internal least privilege is that the scope of a given module or component should only include those system elements that are necessary for its functionality, and that the methods through which the elements are accessed should also be minimal. Layering, modularity and information hiding are constructive techniques for least privilege that can be applied to the internal architecture of the underlying trusted foundation (e.g., separation kernel) to improve the system’s resistance to penetration. The kernel can also be configured to utilize protection mechanisms such as access control and fine-grained execution domains to limit the ability of a subject to perform only the tasks for which it is authorized. In a layered system architecture, enforcement mechanisms of the most critical policies depend on the high assurance layer.

An advantage of RBAC over DAC or BLP models is the isolation of organizational jobs as roles and assignment of minimal permissions to complete a particular job function. Each role then defines a specific set of operations that a user with that role may perform, and a set of objects that the user may access. This provides a level of indirection between subjects and permissions and separates the static time role design from the dynamic nature of obtaining and relinquishing permissions by user and object mappings. This results in a relatively static system design, but one where the individuals filling roles may change much more frequently.

Kuhn, in [43], shows how RBAC can be implemented on a MLS system that uses information flow policies corresponding to a lattice. This is done through the use of a trusted process acting to manage roles for the access control. RBAC on existing MLS demonstrates reuse of current certified and accredited systems, but does not offer the level of information sharing and extensibility required for modern homeland defense.

C. USE CASE ANALYSIS

Unified Modeling Language (UML) was developed by the Object Modeling Group (OMG) and is an international standard used for software development [26]. It is known in its current version (2.0) as a graphical way to specify and construct the artifacts necessary for building a software system.

1. Use Case

Use cases are the generally accepted, standardized method used to represent and present system functionality. The use case describes the system's expected usage in a textual format along with the diagrams. A use case describes a sequence of actions that provide a measurable value for the relationships among actors in a system and are best expressed as an expected use of a system. Use cases are organized using a UML schema. A use case template provides a well-defined method of outlining a system use case and can be used as a good basis for its development. Table 1 depicts the use case template used in this dissertation.

Table 1. Use Case Template

Item	Contents
Use Case Name	Assign a name to the Use Case.
Actors	Name of the actor(s) who participates in the Use Case.
Brief Description	Summarize the Use Case scenario.
Flow of Events	Describe sequentially the basic behavior following this Use Case.
Alternative Flow of Events	For Use Cases, this occupies a partial event in the basic flow. Alternative flow is also meaningful, although in a lesser way. The alternative path is considered when the basic cases are interrupted by a condition in the system or a Misuse Case.
Precondition	Describe conditions and backgrounds that are satisfied before entering the Use Cases and can be ensured by the system itself.
Post Condition	Describe conditions that hold after the Use Case has executed on the system itself.

2. Misuse Case

A Misuse Case represents the actions that a mal-actor should be prevented from performing with respect to the system. The relationships between Use and Misuse Cases can be expressed using relations such as <<*extend*>>, <<*include*>>, <<*prevent*>>, and <<*mitigate*>>. Some instance of misuse can *include* or *extend* a Use Case to achieve undesirable system behavior, while other Misuse Cases show actions *preventing* the Use Cases. The Misuse Case template contains many of the same entities as that for Use Cases and includes a narrative-based textual schema, as well as a Misuse Case diagram. The templates are not unique, as there are many variations among recommended templates for Use Cases and Misuse Cases, but to be able to specify misuse requires examining not only a basic flow, like a Use Case, but also a secondary flow. In other words, we need to examine the Use Case fields and then consider which of these would also be relevant for a Misuse Case template. Fields such as name, actors, description, and flow of events, are relevant to both Use Cases and Misuse Cases. However, misuse cases assume exceptional events that go against standard behaviors of use cases to exploit some element of the system. This requires additional fields to capture the threat involved, the system vulnerability, and the risk of exploitation. Table 2 contains the template used in this dissertation for Misuse Cases.

Table 2. Misuse Case Template

Item	Contents
Misuse Case Name	Assign a name to the Misuse Case.
Actors	Name of the mal-actor who provokes the misuse case.
Brief Description	Summarize a Misuse Case scenario.
Flow of Events	Describe sequentially the basic behavior following this misuse case.
Alternative Flow of Events	For Misuse Cases, this occupies a partial event in the basic flow. Alternative flow is also meaningful, although in a lesser way. The alternative path is considered when the basic Misuse Cases are interrupted by a Use Case.
Precondition	Describe conditions and backgrounds that are satisfied by triggering the Misuse Cases and can be ensured by the system itself.
Assumption	Describe conditions that must be true, but which cannot be guaranteed by the system itself.
Exploited Vulnerability	Describe the vulnerability that exists in the system that is being exploited by the Misuse Case.
Worst Case Threat	Describe the outcome if the misuse succeeds. If the Misuse Case has alternative paths, often this condition will be or contain a disjunction to describe slight variations in outcome.
Capture Guarantee	Describe the outcome guaranteed by whatever prevention path is followed. If no prevention path is followed, one might alternatively formulate a wanted prevention guarantee, expressing what one would want the system to achieve with respect to the attempted misuse, but without stating how.
Related Business Rules	Describe what business rules are violated.
Potential Misuse Profile	Some kinds of misuse are most likely to be performed by intent whereas other may happen accidentally, for example. Some require insiders or people with enormous technical skill, while others do not.
Stakeholders and Threat	This field lists the various stakeholders and their motivations. For misuse cases, this slot is even more important. In this field, risks can simply be described textually.
Scope	This field represents the scope of modeling.

3. Security Use Case

The combination of use and misuse necessitates a Security Use Case to *mitigate*, *detect*, or *prevent* an associated misuse. This Security Use Case represents the system's method of dealing with the vulnerabilities the Misuse Case exploited within the system. The template used in this dissertation for Security Use Cases will model that of the Use Case that was shown in Table 1.

D. SERVICE-ORIENTED ARCHITECTURE

A service-oriented architecture (SOA) is an architecture that supports the discovery, binding, and execution of resources (a.k.a., services) or the composition of resources via a network [27, 28]. Web Services (WS) standards are available for implementing systems.

In this section, we summarize [27] and introduce definitions of SOA, SOA characteristics, and service-orientation principles.

In [27], Contemporary SOA is defined as follows:

> Contemporary SOA represents an open, agile, extensible, federated, composable architecture comprised of autonomous, QoS-capable, vendor diverse, interoperable, discoverable, and potentially reusable services, implemented as Web services.
>
> SOA can establish an abstraction of business logic and technology, resulting in loose coupling between these domains.
>
> SOA is an evolution of past platforms, preserving successful characteristics of traditional architectures, and bringing with it distinct principles that foster service-orientation in support of a service oriented enterprise.
>
> SOA is ideally standardized throughout an enterprise, but achieving this state requires a planned transition and support of a still evolving technology set.

The preceding definition on contemporary SOA is based on separation of concerns [27]. Services encapsulate logic for solving the decomposed individual concerns of existing complex problems.

There are three basic components of the SOA architecture: services, description, and messaging. The service is the executable code; the (service) description contains the name of the service, location of the service, and the input and output exchange requirements; and messages are independent units of communication that the services use to communicate [27]. An adaptation from [27] shows these components in Figure 5. These components could also describe a distributed architecture; yet SOA is highlighted by how each of these components is designed; using service-orientation principles.

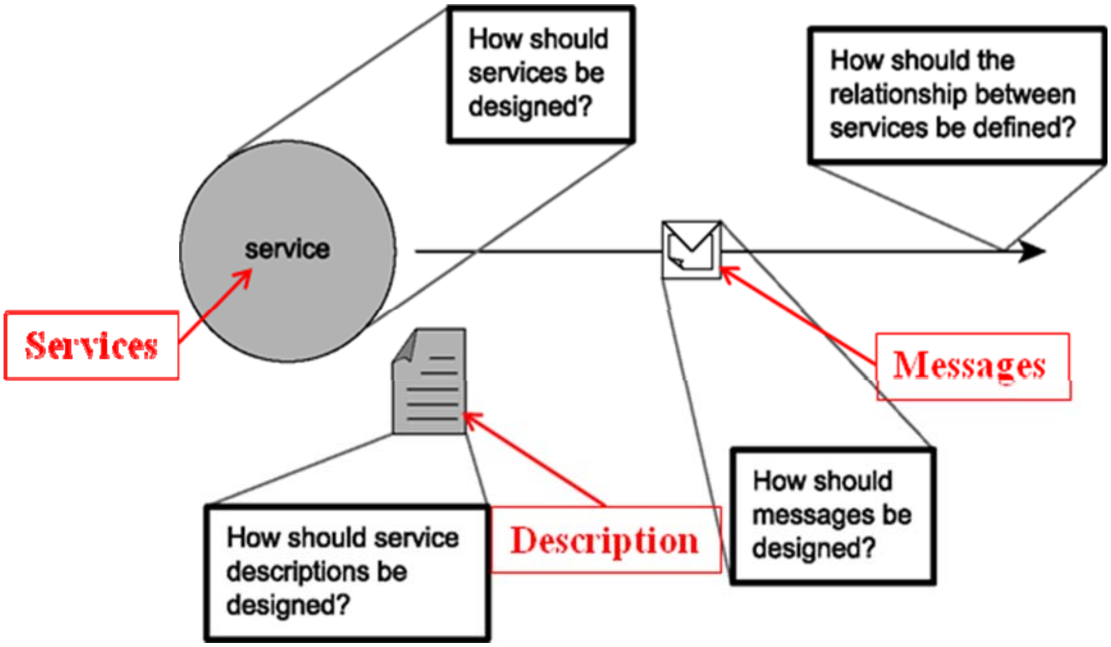

How should services be designed?
How should the relationship between services be defined?
service
Services
Messages
How should service descriptions be designed?
Description
How should messages be designed?

Figure 5. Basic SOA Components and Design Relation, from [27]

Service-orientation principles are “a set of principles most associated with service-orientation” [27]. These principles are applied to the development of the basic SOA components. The eight principles are listed in Table 3.

Table 3. Service-Orientation Guiding Principles, from [27]

Service-orientation principle	Brief description
Reusability	Services are designed to support immediate and potential reuse
Service contract	Services are designed with formal contracts which describe the service and expose a services data sharing requirements
Loose coupling	Services are designed to relate without dependencies on other services
Abstraction	Service contracts are the only visible entity of a service. The actual service is of no concern to the user
Service-orientation principle	**Brief description**
Composability	Services can make up other services
Autonomy	Services are designed to be independent, self-governing within an explicit boundary
Statelessness	Services are designed so as not to manage state information
Discoverability	Services are designed to be discovered for use; they expose their formal contract for anyone to use

One additional piece of the primitive SOA definition is what [27] calls the implementation platform. This is where Web Services are used to integrate the components and provide our service-oriented solution.

Contemporary SOA is based on primitive SOA, but differs in that Primitive SOA represents what can and is being done with existing Web services technology rather than what is being done with current Web Services technology and what can be done in the future with extensions to the current WS. Table 4 lists the common characteristics of Contemporary SOA, as described in [27].

Table 4. Common Characteristics of Contemporary SOA, after [27]

Common Characteristics of Contemporary SOA
Contemporary SOA is at the core of the service-oriented computing platform
Contemporary SOA increases Quality of Services (QoS)
Contemporary SOA is fundamentally autonomous
Contemporary SOA is based on open standards
Contemporary SOA supports vendor diversity
Contemporary SOA fosters intrinsic interoperability
Contemporary SOA promotes discovery
Contemporary SOA promotes federation
Contemporary SOA promotes architectural composability
Contemporary SOA fosters inherent reusability
Contemporary SOA emphasizes extensibility
Contemporary SOA supports a service-oriented business modeling paradigm
Contemporary SOA implements layers of abstraction
Contemporary SOA promotes loose coupling throughout the enterprise
Contemporary SOA promotes organizational agility
Contemporary SOA is a building block
Contemporary SOA is an evolution
Contemporary SOA is still maturing
Contemporary SOA is an achievable ideal

A full description of each characteristic listed in Table 4 can be found in [27]. This list is used to show that Contemporary SOA is neither merely Web Services and service-oriented principles, nor is it a commercial off-the-shelf (COTS) product that provides a guaranteed solution. It is a highly adaptable reference point to begin creation of a flexible system architecture that relies on services. The eXtensible Markup Language (XML) is the basis for much of the web services. XML Schemas are used to describe rules for an XML document to conform to and to provide validity through the use of an XML Schema Definition (XSD). The XSD is an instance defining a document by constraints on elements, attributes, relationships, and data.

Web Services Description Language (WSDL), SOAP (formerly Simple Object Access Protocol), and Universal Description, Discovery, and Integration (UDDI) are commonly referred to as the first generation Web services standards [27, 28]. Each of these standards represents a concern for the development of a Web service. WSDL is a standard used to develop an XML-based document that contains, at a minimum, the service name, location, and input and output requirements. This is the contract for a service and defines services as ports or network endpoints. The WSDL document is a user's interface to an actual service and the information in a WSDL resides in a UDDI registry so that the service can be discovered. UDDI is a specification used to design an XML-based registry service for Web services. The information contained in a WSDL has a standard format, as outlined in the OASIS UDDI specification, and mapped to a UDDI data model. The UDDI is open to SOAP messaging queries, which allows for any registered service to be found and also describes the protocols and messaging formats necessary to communicate with the service in question. The communications framework is further described by the SOAP standard. SOAP is an XML-based language and a platform-independent communications protocol for exchanging messages between services over a network. SOAP is a stateless, one-way message exchange that is typically transported via HTTP/HTTPS or SMTP. A graphical representation of the core standards and how they relate is provided in Figure 6.

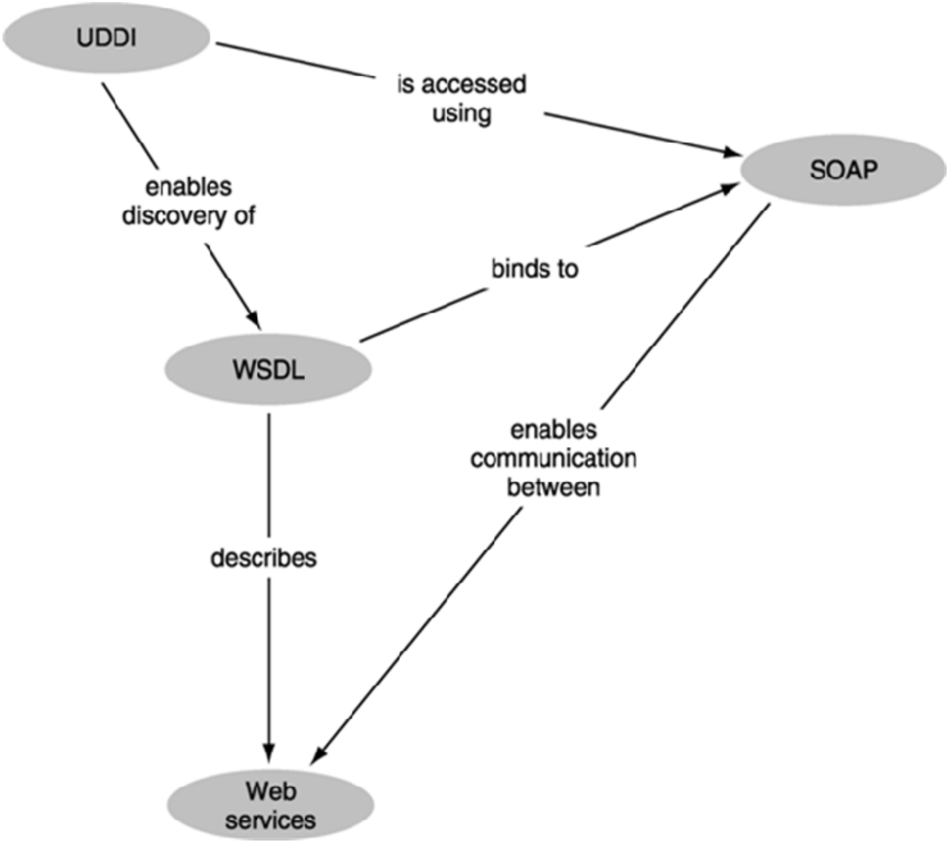


Figure 6. Basic SOA with Core Web Service Standards, from [27]

Additionally, Web Services Business Process Execution Language (WS-BPEL) provides a means for specifying how to integrate elements. It is also used to import and export functionality by WS interfaces. This is intended to model abstract and executable processes where BPEL can be used as an orchestration language, so that the executable process and message exchange are controlled (central control of a distributed system's behavior). This is different than choreography, where protocols of interaction and legal sequences of messages for interoperability are specified (a distributed system without centralized control).

E. INFORMATION BROKER

The Information Broker (IB) is a key service in a SOA implementation of a MLS system. It enforces access control policy and provides for the orchestration of security services. Turner et al. discuss, in [44], the creation and application of an IB in SOA-based system in a healthcare domain. This is a single security level, but does encompass multiple domains and distributed data repositories, similar to this research. Turner et al. use XACL as the specification language for the IB [44]. The IB can be likened to an Enterprise Service Bus (ESB) that provides messaging between the components or business logic [29]. The ESB, however, is not designed for a true SOA system to provide a service, but rather to interface via messaging between two dissimilar services or otherwise incompatible components. In contrast, the IB serves as the orchestration mechanism at the heart of the trusted computing base (TCB) and is replicated for each confidentiality level.

F. MULTILEVEL SECURITY

Multilevel security is used to describe systems that can operate at varying security levels without the need for separate hardware to provide the access to a secondary classification level of information. Two major categories of implementation exist for these systems; those utilizing a separation kernel and those without. Rushby offers that the role of a security kernel is to separate environments on a single processor as if they were running on physically separate machines [30]. Typically, it was the "one component that partitioned many kinds of resources (complex implementation), and either enforced a single operational security property (too rigid to be useful) or several (too complicated to be credible)" [31]. Several foundational elements are required to include: the ability to prove no information flow channels exist between domains, non-bypassable, tamperproof, and being small enough to test and analyze completely.

1. Separation Kernel Based

Multiple Independent Levels of Security (MILS) is a high-assurance security architecture based on processing separation and controlled information flow. MILS allows for independent evaluation of security components and trusted composition. Much work has been done with MILS by VanFleet [8] as an extension of the design of secure systems from Rushby [30]. This is fundamentally different than the SOA based implementation. MILS focuses on multiple levels of security on a single system and not via distributed computing. MILS systems could serve as endpoints offering multiple sensitivity levels from a single source to augment the approach discussed in this research.

While there is a lack—primarily due to cost—of proprietary systems that have been built, certified and accredited, the door has opened for the use of non-developmental items. Boeing is now using the WindRiver VxWorks MILS Platform 2.0 separation kernel for its embedded real-time systems. Additional and comparable software separation kernels to support MILS are offered by Green Hills (Integrity 178B) [32], and LynuxWorks (LynxSecure) [33]. These efforts have resulted in EAL6+ certified systems with additional examples from Rockwell Collins' AMP7, and Boeing's Secure Network Sever that with additional integration and effort could result in a full implementation. Cisco, Microsoft, EMC and Decru partnered to create the Secure Information Sharing Architecture (SISA). SISA utilizes commercially available products to provide information sharing and separation controls at the PL3 (Medium Assurance) level [34]. SISA does not support SOA and is not a high assurance solution.

2. Non-separation Kernel Based

BAE Systems' STOP OS is an example of a non-separation kernel based implementation for MLS [35]. BAE integrates the STOP OS with their Next Generation XTS Guard into a Secure Application Platform as the basis for specification and enforcement of security policy in MLS, cross-domain environments [36-38]. This approach is heavily reliant on the use of the XTS Guard and the security controls enforced by the operating system.

The information broker approach via SOA that is described in this research is based on the services and the distributed environment inherent with the model. While the underlying individual operating system is independent of the information broker's mission, the STOP OS is designed as a general purpose OS to be run on individual systems to maintain data separation. The Radiant Alloy model provides for the TLS and IB to make the separation decisions prior to the information arriving at an individual system. This is an OS that could be used in conjunction with Radiant Alloy and the IB, but could not replace the TCB and functionality of the IB itself that we are specifying policy rules to implement. The functions of the operating system to control access to the hardware are not an implementation goal of Radiant Alloy, where any network capable device is inclusive of the intended platforms for use.

McCullough, in [41], describes a security property for MLS systems called restrictiveness. This property refers to the ability of a user to infer sensitive information from the system within the scope of the security policy. Additionally, McCullough offers this restrictiveness as holding with a hook-up or composable property when combining secure systems to a single composite system.

Goguen and Meseguer, in [42], present verification of an MLS keying on security policy satisfaction for a specified security model. Their approach analyzes non-interference of information flows for the security policy and the high-level specification to verify the security of a system. This work did not encompass inference or information aggregation creating security policy violations as we do in this dissertation.

Trusted Solaris provides an implementation of RBAC and supports MLS. Trusted Solaris, however, is designed for the operating system processes only and not for the movement and dissemination of data in a distributed or SOA system.

Osborn, Sandhu and Munawer in [39], attempted to show the inclusion of MLS policy (BLP) using RBAC roles. They developed models for using RBAC to encompass both DAC and MAC policies via roles. The use of a lattice-based access control model and the generality of the RBAC model allows for the simulation of these other models. Despite RBAC being more expressive than BLP and allowing the limited modeling of MLS within a role, the implementation in RBAC relied on a single administrator role. This lacks the true separation of duties and responsibility necessary for a MLS system. This would involve a violation of the *-property, and Simple Security property. RBAC was defined in [40], then expanded in [41] before its acceptance by The National Institute of Standards and Technology (NIST) in [42]. RBAC has also been shown in [25, 43-48] to provide a sound platform to develop assurances of access control using administrative roles and varying access control mechanisms. Additionally, Kuhn provided a concept to provide RBAC on an existing MLS system in [43].

Freeman, Neely, and Heckard [14] offer a way to map security policy into a model based on the security requirements and the system architecture. Their Boundary Flow Modeling (BFM) approach helps to model the security policy with regard to the system architecture it will operate in, rather than trying to enforce a generic policy on an incompatible domain. It also allows for mathematical formalization of the policy model as an additional problem with a distributed system is that in general they are non-deterministic, since separate processors in the system execute state changes in an order that is not predictable relative to the others in the system. But, the security requirements must still be adhered to. Boundary flow modeling addresses this by defining relationships rather than functions, between inputs and outputs where these relationships can be modeled mathematically as a relation and can account for nondeterministic behavior in the system [14]. Their BFM method is designed to aid in the development process and does not provide an implementation method as addressed in this research.

3. Other Multilevel Security Work

Multilevel security is a common desire with information repositories. The Sea View security model [60] addresses the concept of multiple levels enforced by a reference monitor and the extension of individual data classifications in the database for enforcement and creates multiple database servers for every level the user dominates.

This approach does not allow for the capability of connecting to multiple disparate data stores outside of a relational database, nor does it account for the prioritization and emergence of additional data flows that must be accounted for within current operational systems.

G. GUARDS

Guards are mechanisms designed to limit the exchange of information between systems [49] and utilize any number of inputs to determine the release or modification of the information in question. This is a critical element of any cross-domain effort and its implementation is usually the hallmark of the system architecture itself. There is much prior work related to differing guard implementations, yet none of these allow for the same level of expressiveness and granularity as provided through the integration of RuleML and the Information Broker described in this research.

The Information Support Server Environment (ISSE) Guard from the Air Force Research Laboratory (AFRL) allows for bi-directional email messaging, imagery, and Microsoft Office file transfer between interconnected domains [50]. This guard supports a single high-side system with up to eight low-side systems, but it does not provide a means for the system to ensure that labels a user associates with information provided to the system are consistent with the sensitivity levels that the user is allowed to access. This is not as scalable as the SOA-based implementation of Radiant Alloy and the guard characteristics are defined for a limited scope of transfer [51]. The National Security Agency (NSA) and AFRL are jointly working to integrate XML capabilities into the ISSE to allow for transfer between domains and provide this via web services. This is different than the use of RuleML with an information broker and is content based rather than configuration based. The ability and the need to transfer data across the domains using the ISSE, as well as future expansion to support new XML capabilities, is the goal.

The Defense Messaging System (DMS) is a follow-on cross-domain messaging service created from the legacy Defense Switched Network (DSN) AUTODIN System based on labels. DMS provides two categories of service: High Grade Service (HGS) uses modified commercial email clients and a FORTEZZA token-based Class 4 Public

Key Infrastructure (PKI) certificate, while Medium Grade Service (MGS) uses commercial email clients with a Class 3 PKI certificate [52]. The DMS supports four DMS security domains: Unclassified (U), Secret (S), Top Secret–Collateral (TS) and Top Secret/Sensitive Compartmented Information (SCI); according to the transport network. DMS utilizes a High Assurance Guard (HAG) for cross-domain exchange of message traffic, which is different than the use of an information broker as the central control element between data repositories. The idea of disparate classifications using a message system is limited in the ability of the guard to check the content and parse the messages.

The Navy Research Laboratory (NRL) modified its original Pump to a Network Pump to create a high assurance guard that supports cross-domain messaging [53]. This allows transfer from low to high and prevents the downward flow, but also provides an acknowledgement indication of the transfer back to the low-side. This approach also relies on modifying the timings of the response messaging in order to mitigate the use of transfer responses as a covert channel. This approach has a few drawbacks compared with the use of an Information Broker and specification of policy using RuleML. Although offering limited support of information sharing among disparate users, the types of transfer are limited while the use of RuleML specifying a security policy allows for greater expressivity and can also be constructed to eliminate the inference ability of a lower classification user. This also allows for multi-level classification of information and thus, a different view related to similar objects between users (e.g., a ship's detailed track history). The NRL Pump also is intended as a part of a larger collection of networks rather than as an information destination connected via a service-oriented architecture and rapidly scalable via a replication of IB service and is only a one-way guard as opposed to bi-directional.

The Monterey Security Architecture (MYSEA) is a related concept to allow for distributed information sharing for MLS [54]. However, the MYSEA is an implementation of the Trusted Computing Base (TCB), and would serve as a viable alternative to Radiant Alloy as used in this research. The MYSEA does not implement a defined ruleset supporting policy, but rather the environment in which to create the enforcement and authorization mechanisms for the multilevel security policy.

BAE Systems uses the Next Generation XTS Guard as an integral piece of their Secure Application Platform, along with the STOP OS described previously [36-38]. The XTS Guard is designed for use as a gateway for desktop, server, or network environments. It is designed to separate data from varying security levels based on label and sensitivity, yet it does not allow for the openly configurable rulesets and must rely on proprietary hardware and software elements to provide its services.

H. FIREWALL AND IPS LANGUAGES

Current enterprise-level firewalls and Intrusion Protection Systems (IPS) have reached a level of sophistication that was not available in prior generations of devices. The prior reliance on port number, protocol, and IP address filtering have been superseded by a more dynamic ability to restrict or block traffic based on triggers and performance thresholds established for the device. While the functionality and concept of creating and running rules to support filtering and blocking of network traffic by a firewall and intrusion protection system (IPS) are similar, the firewall and IPS rules are much different than the use of RuleML in this research. Typical configuration of these network appliances is via a graphical user interface (GUI) and is proprietary to each respective vendor. However, these are all similar in the underlying rule generation, which is very simple logic. This allows for the devices to operate at very high data rates to perform the specific function of filtering content in the network, but does not permit complex logic such as a filtering of a suspicious domain based on recent network traffic patterns. These firewall rules are executed in a simple beginning to end list, where the first rule is evaluated and if it is not triggered the execution continues to the next rule in sequence [55]. As soon as a rule is triggered, the remaining rules are ignored and the initial decision is upheld for that particular rule. This prevents the ability to ensure rule consistency throughout the rule list and is one of the primary concerns, from both network appliance vendors and enterprise clients, as rulesets for these devices become ever larger to account for growing enterprise needs and emerging threats.

Multiple vendors offer network appliance solutions, such as firewalls (FW), intrusion protection systems (IPS), and intrusion detection systems (IDS). Each of these has a variation with its own corporate branding and a different user interface, yet the underlying structure and execution models are similar. Cisco Systems Inc. offers the ASA 5500 Adaptive Security Appliance [56] line of firewalls as the top-tier of their efforts. Check Point Software Technologies Ltd. offers the 61000 Security System [57]. McAfee Firewall Enterprise [58] is the top offering of McAfee and is branded as a next-generation device. While each of these vendors touts advanced security and adaptive measures for filtering and content control, the underlying operation is still reliant on a simple logic that lacks complexity and expressiveness. Overcoming these weaknesses, RuleML in our example allows for the use of chaining rules and can be checked for consistency with the ruleset. The ability to add a complex rule is also evident in RuleML. However, while this functions well for a specific enterprise device, it is not expressive enough and only covers a single domain when compared to the use of RuleML and an Information Broker as we describe

In line with the network appliance is the functionality offered by the IPS, and how it processes network flow. One of the most widely used of these, offering both an open source rule engine and rule language, is Snort. Snort allows you to extend its predefined syntax and rules to meet the needs of the network. This is a common application within a corporate enterprise network and is used on many different hardware appliances to support intrusion detection and prevention. A Snort rule can be broken down into two basic parts, the rule header and options for the rule. The rule header contains the action to perform, the protocol that the rule applies to, and the source and destination addresses and ports. The rule options allow you to create a descriptive message to associate with the rule, as well as check a variety of other packet attributes by making use of Snort's extensive library of plug-ins. The generic Snort rule is: *action protocol src_ip src_port direction dst_ip dst_port (options)* [59]. This is again simple logic, not allowing for rule chaining and lacking in expression for more complex filtering. When a packet comes in, its source and destination IP addresses and ports are compared to the rules in the ruleset. If any of them is applicable to the packet, then the options are compared to the packet. If all of these comparisons return a match, then the specified action is taken. All other rules in the set are excluded from consideration after a single rule is triggered. As we can see, the underlying basis is simple and designed for speed of execution for high-flow network entry points where throughput is paramount. Yet, these rules and the language used to support them are not sufficient to meet the needs of our cross-domain information broker.

I. XACML

eXtensible Access Control Markup Language (XACML) is an OASIS standard. XACML is the access controller of the Web Services languages and the current reference implementation has a single policy decision point (PDP) and a policy enforcement point (PEP).

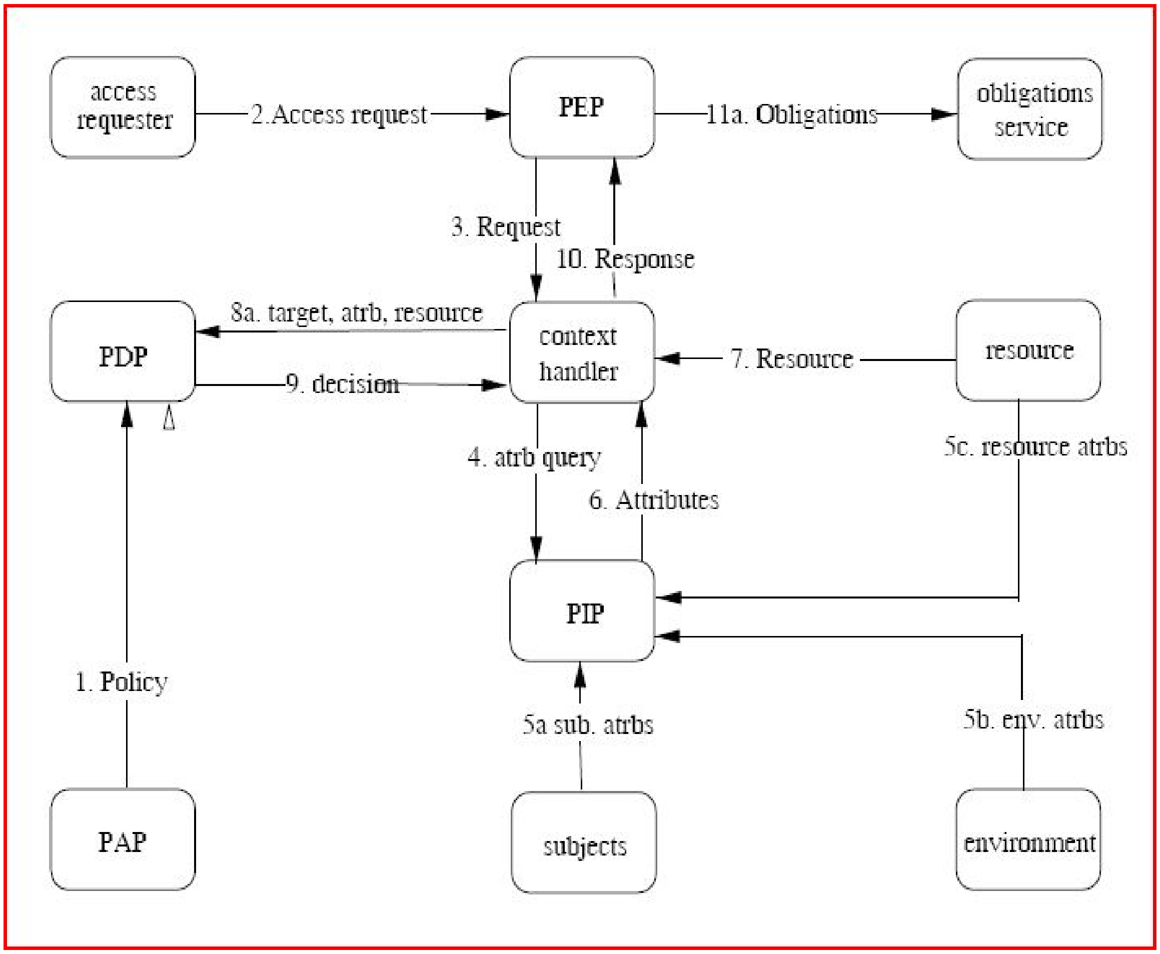


Figure 7. XACML Architecture, from [60]

The current XACML specification has three main entities as shown in Figure 7. The main components are:

1. **Policy Administration Point (PAP):** Entity that creates policies or policy sets.
2. **Policy Decision Point (PDP):** Entity that evaluates applicable policy and renders an authorization decision. The answer given by the PDP is one of (1) *permit*, (2) *deny*, (3) *insufficient information to decide* or (4) *error*, occurred in the execution.
3. **Policy Enforcement Point (PEP):** Entity that performs access control by enforcing authorization decisions.

Figure 7 also shows the dataflow of the XACML reference implementation. First, the PAP creates a policy. At request time, an access request arrives at the PEP (flow 2), and is sent to the context handler (flow 3). The context handler determines resources to be accessed and attributes of the requester, resource and the environment, collects all required attributes and forwards them to the PDP (flows 4 through 8). The PDP then acquires the policy from PAP (flow 1), evaluates the relevant policy and relays the decision (flows 9, 10) to the PEP through the context handler, which then enforces the authorization decision [60].

The policy syntax (XML) includes language constructs to identify the *resource*, the *action* (to be performed on the resource), the *subject*, and *constraints* on the access. In XACML parlance, this collection of entities is called a *target*. The request syntax (XML) identifies the resource, the action, the subject. The decision engine (PDP) *matches* the incoming request to available policies to discover all applicable policies. If more than one policy is applicable, then the PDP uses a *policy-combination algorithm* [61] to determine the evaluation result. In essence, the combination algorithm states how to combine the result of each applicable policy. For example, it can state that the final result is the conjunction/disjunction of the individual results, or the first-applicable policy evaluation result is the final result [60].

J. RULEML

Rule Markup Language (RuleML) specifies Prolog-like rules that use an XML-like syntax, and consequently are valuable with SOA. RuleML was initially developed by the Rule Markup Initiative to express rules in XML for various tasks. The semantic foundation of RuleML is based on datalog, which combines SQL and Prolog, and can be considered as a subset of logic programming [62]. RuleML serves the Resource Description Framework (RDF) as a canonical Web language. While RDF is the basis for the larger data interchange within the web, RuleML covers the entire rule spectrum, from derivation rules to transformation rules to reaction rules. RuleML can be used to specify queries and inferences in Web ontologies, mappings between Web ontologies, and dynamic Web behaviors of workflows, services, and agents [10].

The RuleML package provides a namespace for XML, facilitating reuse. Top-down or bottom-up rules can be used, specifying deductive logic, rewriting, and inference. Rules can be stated in natural language, some formal notation (e.g., Backus-Naur form), or in a combination of both. The combination of natural language and formal notation offers the most nearly universal appeal to permit Web-based rule storage, interchange, retrieval, and application.

The RuleML namespace has a hierarchy of rules that consists of varying categories: reaction, transformation, derivation, facts, queries, and integrity constraints. The hierarchy is shown in Figure 8, where two main categories of reaction rules and transformation rules form the basis for all other categories of rules.

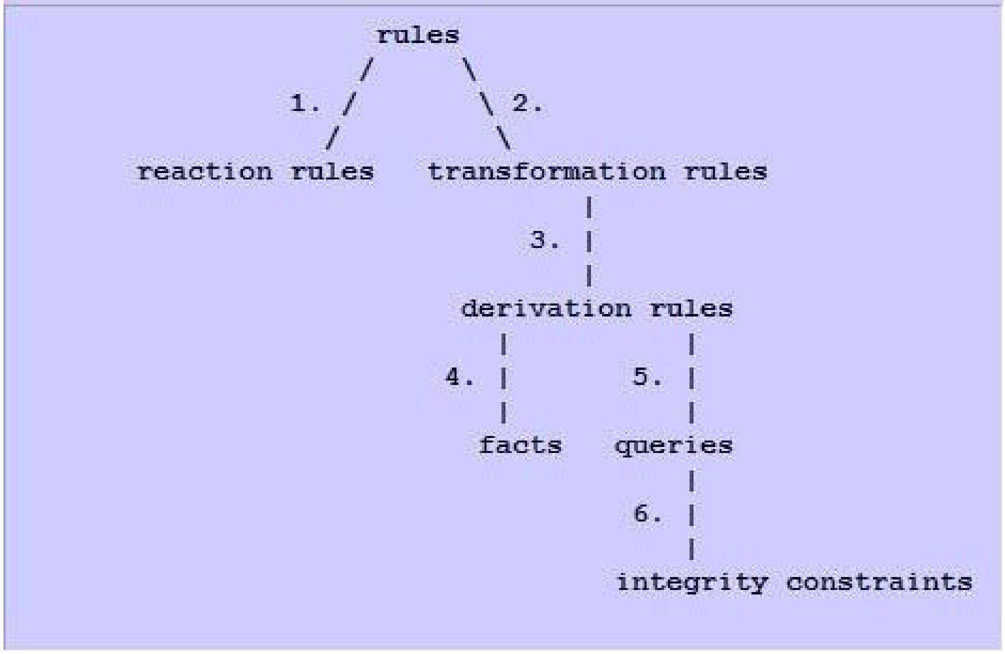


Figure 8. RuleML Hierarchy, from [10]

While general rules, as the all-encompassing rule category, could implement all other categories of rules, special-purpose syntaxes for each of the subcategories were created to allow for ease of refinement and application. The following show basic markup syntaxes for each of the various categories:

- Reaction rules: **`<react> <_event>`** *trigger* **`</_event> <_body> <and>`** *prem1 ... premN* **`</and> </_body> <_head>`** *action* **`</_head> </react>`** reducible to **`<rule> <_event>`** *trigger* **`</_event> <_body> <and>`** *prem1 ... premN* **`</and> </_body> <_head>`** *action* **`</_head> <_foot>`** *empty* **`</_foot> </rule>`**
- Transformation rules: **`<trans> <_head>`** *conc* **`</_head> <_body> <and>`** *prem1 ... premN* **`</and> </_body> <_foot>`** *value* **`</_foot> </trans>`** reducible to **`<rule> <_event>`** *active* **`</_event> <_body> <and>`** *prem1 ... premN* **`</and> </_body> <_head>`** *conc* **`</_head> <_foot>`** *value* **`</_foot> </rule>`**
- Derivation rules: **`<imp> <_head>`** *conc* **`</_head> <_body> <and>`** *prem1 ... premN* **`</and> </_body> </imp>`** reducible to **`<trans> <_head>`** *conc* **`</_head> <_body> <and>`** *prem1 ... premN* **`</and> </_body> <_foot> true </_foot> </trans>`**

- Facts: `<fact> <_head>` *conc* `</_head> </fact>` reducible to `<imp> <_head>` *conc* `</_head> <_body> <and> </and> </_body> </imp>`
- Queries: `<query> <_body> <and>` *prem1 ... premN* `</and> </_body> </query>` reducible to `<imp> <_body> <and>` *prem1 ... premN* `</and> </_body> <_head>` *bindings(var1, ..., varK)* `</_head> </imp>`
- Integrity constraints: `<ic> <_body> <and>` *prem1 ... premN* `</and> </_body> </ic>` reducible to `<query kind="closed"> <_body> <and>` *prem1 ... premN* `</and> </_body> </query>`

Reaction rules incorporate various production, action, reaction, complex event notification, event messaging and temporal or action logic rules. These rules can be reduced to general rules that return no value and these rules can only be applied in the forward direction in a natural fashion, checking (observing) for events (conditions) and performing an action if and when all events have been recognized (fulfilled). Reaction rules allow for event notification and messaging between services, as well as temporal and state-based logic rules. These types of rules still provide structure to accommodate a wide range of business cases in their expressiveness. Production rules are action rules where a condition is met, the IF, and an action is taken, the DO. Action rules consist of triggers, the ON, resulting in the action, the DO. Overall, reaction rules are most often exemplified as logic rules like those used in a firewall configuration. The forward chaining of rules is popular with expert systems and production rule systems. Forward chaining uses data points against the ruleset to infer additional information, as the data is compared to the existing rules. By checking the antecedent or IF clause of a rule, the consequent or THEN can be inferred. These conclusions then add further data points to use against the ruleset until an endstate or goal clause is reached. Overall, forward chaining ends with a result based on the original data and additional data points arriving from a changing situation can quickly be addressed by an existing ruleset to gain additional inferences and realize a new endstate.

In contrast, the backward direction is normally preferred for transformation rules. The category of rules can be reduced to general rules whose event trigger is always activated. Backward chaining begins with a result and seeks to find rules that will support the endstate. Each of the consequents, or IF statements, that are required to reach the goal endstate are added to the inference chain as items to resolve and additional goals to match. This method is used in automated theorem provers and expert systems and uses the goals to determine which rules become active, as opposed to checking the rules with the data given and inferred as in the forward chaining process.

Derivation rules can be reduced to transformation rules that on *success* return *true*. They can be applied in the forward direction or in the backward direction equally. The backward direction reduces the proof of a goal (conclusion) to proofs of all its sub-goals (premises). Since in different situations different application directions of derivation rules may be optimal (forward, backward, or mixed), RuleML does not prescribe or restrict any one of these. Facts can be reduced to derivation rules that have an empty conjunction of premises (*true*), and facts or *unit clauses* have no applied direction. Derivation rules are based on reasoning and typically expressed as an IF-THEN relationship.

RuleML also supports the use of queries within the basic code structure. These queries can be reduced (transformed) into derivation rules. Each query transformed to a derivation rule will have a *false* conclusion. Queries also are applied as top-down goals and can be proven backwards; but they can also be proved forward via goal-directed bottom-up processing. Integrity constraints can be reduced to queries that produce no variable bindings (closed), and are usually forward-oriented (i.e., triggered by update events). However, integrity constraints can also be backward-oriented, to show (in)consistency by fulfilling certain conditions (without recognizing an event).

In this research, reaction rules will be used to provide a forward chaining path to demonstrate the viability of specifying a security policy for the information broker. The overall ruleset is simplified to account for a limited number of cases as presented in this work. The use of more sophisticated rulesets and using other types of RuleML rules, like derivation and transformational rules, is not explored but is possible with this specification.

THIS PAGE INTENTIONALLY LEFT BLANK

III. OPERATIONAL CONTEXT

The operational context of our study is the Navy Tactical Exploitation of National Capabilities (TENCAP) Program's prototype information brokering system named Radiant Alloy. Radiant Alloy is a service-oriented, multilevel secure, enterprise information system designed to provide confidentiality for users of varying security levels, and data stores of various security levels over heterogeneous domains. Radiant Alloy provides a trusted means for sharing both unclassified and sensitive data among diverse user levels and domains, while maintaining anonymity of the data source, as depicted in Figure 9.

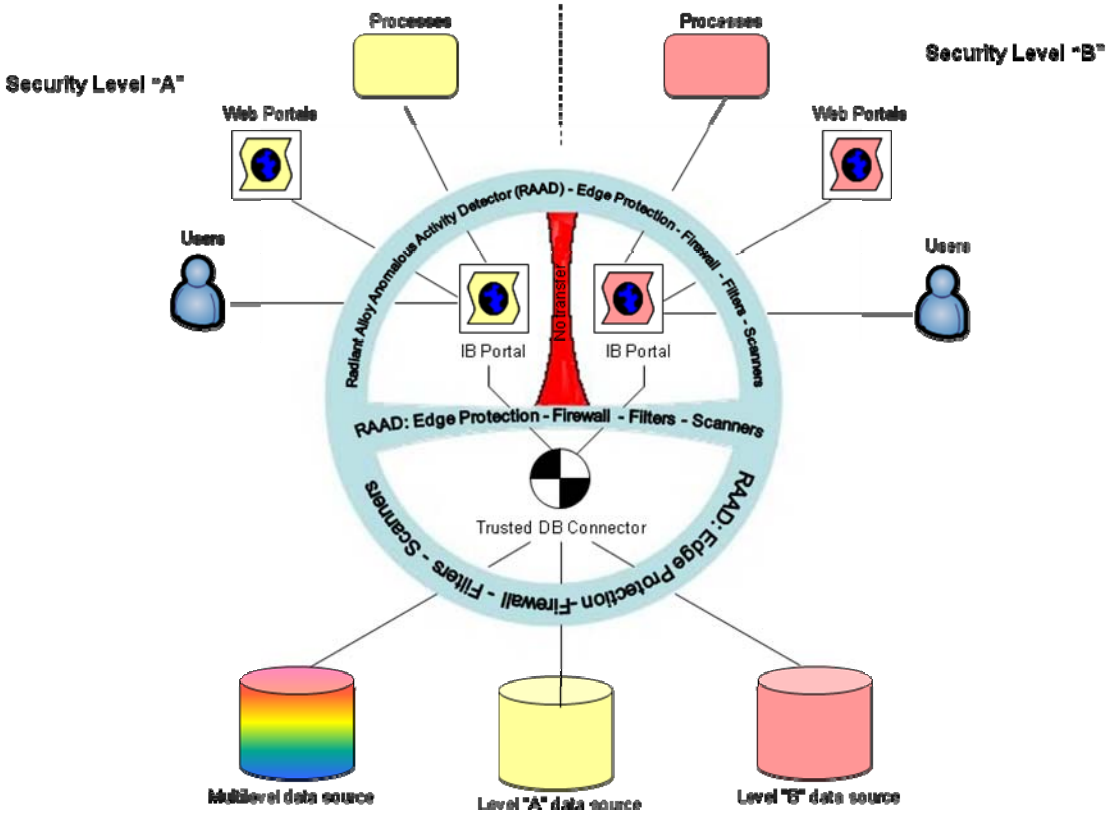

Processes
Processes
Security Level "A"
Security Level "B"
Web Portals
Web Portals
Users
Users
Radiant Alloy Anomalous Activity Detector (RAAD) - Edge Protection - Firewall - Filters - Scanners
No transfer
IB Portal
IB Portal
RAAD: Edge Protection - Firewall - Filters - Scanners
Trusted DB Connector
RAAD: Edge Protection - Firewall - Filters - Scanners
Multilevel data source
Level "A" data source
Level "B" data source

Figure 9. Radiant Alloy High Level Concept, from [7]

Two separate security levels (A and B) are shown, along with the system's intention to prevent interaction between the levels during its use, indicated by the "No Transfer" bar. The information broker (IB) resides behind firewalls and network edge protection, yet is accessible using web services (WS) for a mixed MLS-RBAC model where MLS will divide varying levels of systems such as JWICS (Top Secret), SIPRnet (Secret) and NIPRnet (Unclassified). RBAC will be used within each domain to enforce further control of access (i.e., finer granularity) based on the roles of individual users. The IB is also expected to prevent inadvertent disclosure of information between security domains and facilitate authorized sharing (i.e., answer valid queries) among classification domains. This is a critical capability of interest to both the Homeland Defense and Security communities. The IB must also allow authorized transfer and downgrading of data for users of differing domains. The Trusted Data Connector, shown in Figure 9, is the plug-in point for the data repositories and serves as the trusted service within the system. Our Use Cases and Misuse Cases capture example scenarios in which the information is needed to be shared between different domains and the inferences that need to be prevented while sharing this information.

Data accesses are controlled by the IB core and the trusted data connector for each security level, where the IB is replicated. In a MLS system, the IB must be replicated to allow for separation of the domains. However, this adds complexity to the system since the IB must now maintain consistency between implementations, in addition to isolating users from the data sources. The IB must also support global and local instantiations for each of its domains. The localized version will allow for finer and more restrictive control at a local level, but must be integrated and verified against the global IB for access requests. While the local IB policy must be equal to or more restrictive than the global IB for accesses, it must also verify requests with the global version to enforce the more restrictive control; that is, the local version cannot give more access, it can only lessen the level of access. This would also account for changes made at the global IB that may not yet be reflected in a localized instantiation of the IB for that security domain. Each IB will have a separate Policy Decision Point (PDP) to evaluate access requests. This local decision must be sent to the master or global PDP to combine the access decision into a finalized context. The IB must also allow for a user to have write access to a lower level data store (which is explicitly forbidden in the BLP model), as well as preventing leakage from different domains. The information broker must enforce a mixed model access controller to provide the separation of users and data necessary to protect and enforce the confidentiality, integrity, and availability in a MLS system, as opposed to a monolithic security kernel that brokers all accesses and controls all resources. The IB also serves as a request broker for use with many different back-end data sources. The IB must ensure that all requests are executed within the databases at the security level at which requests are made and limits the results based on the clearance of the requesting user or the classification level of the network connection whichever is more restrictive. The user is authenticated to the IB and not to the data source directly, thus never obtaining permissions for the true data-store object. This provides anonymity and integrity of the source, as well as prevents users from undesired (from a security viewpoint) file operations [63].

Trusted sharing among classification domains is a critical capability of interest to both the Homeland Defense and Homeland Security communities. The IB must also allow authorized transference and downgrading of data for users of differing domains. Radiant Alloy will provide a means of secure[14] and trusted[15] information sharing that is currently non-existent at the enterprise level across communities of interest. The ideas of a secure system and a trusted system lead us to the certification and accreditation process, where the trusted system is one that meets “well defined requirements under an evaluation by a credible body of experts who are certified to assign trust ratings to evaluated products and systems” [3]. This process allows us to assign a level of trust to the architecture, implementation, life cycle and management, and disposal of the system to meet the information assurance requirements necessary for the intended application. As the level of trust increases, so do the number of requirements and the stringency of

[14] Secure sharing refers to the ability of the system to protect the confidentiality of information from inappropriate access across the varying security domains [3].

[15] Trusted sharing refers to the level of confidence or belief that users have in the ability of the system to protect their information and resources across those same domains to be safe from compromise [3].

those requirements. Ultimately, we cannot guarantee a system to be secure but we can assert a level of trust to the protections and security mechanisms[16] the system affords. This includes the implemented functionality, as well as the assurance that the functionality is correct. No SOA system has been accredited at a High Assurance level. One of the reasons for the overall lack of certified and accredited systems is that the architectural implementation and the security policies that we are trying to enforce within that system are difficult to implement, while maintaining the viability of the original system intent. Security properties, and the formal models associated with them, can be used to gain improvements in effectiveness, efficiency and correctness of a system's security properties. But, bridging the gap between security requirements and the security mechanisms used in an implementation is very complex and expensive. When designing a system architecture to support a foundational requirement of security, we often lose capabilities in other areas, such as usability or performance. This has become a stumbling block in creating and fielding systems, because if they meet the requirements for High Assurance they often sacrifice in other areas and might not meet the user needs. Navy (TENCAP) is attempting to build a service-oriented architecture that supports MLS. Figure 10 depicts the proposed system architecture for Radiant Alloy at the high-assurance level. This architecture supports multiple user bases, multiple data stores, and industry-standard protocols for implementation, through the use of a mixed model access control (MMAC) policy. In the MDA context, Radiant Alloy offers the ability of an Unclassified user to obtain data from a data store, while a Secret user is obtaining a similar data feed (but with more detail commensurate with the classification) from the same data source. It becomes a significant problem, however, if a lower level user can observe any service delays when the higher level user is being served by the system. This then becomes a covert channel, which is contrary to one of the primary objectives of an MLS system, that being preventing creation of covert channels. This becomes not only a challenge to designing and implementing a system, but also to the certification process to ensure that it is free of covert channels or that the channels have very limited bandwidth

[16] Security Mechanism is a method, tool, or procedure for enforcing a security policy [3].

where the risk of exploit can be accepted by the system owner. In a shared resource system, it becomes impossible to remove all the covert channels. This ability to share data, without attribution to the source, and to multiple users among differing domains concurrently, would serve as a true combat multiplier for our military.

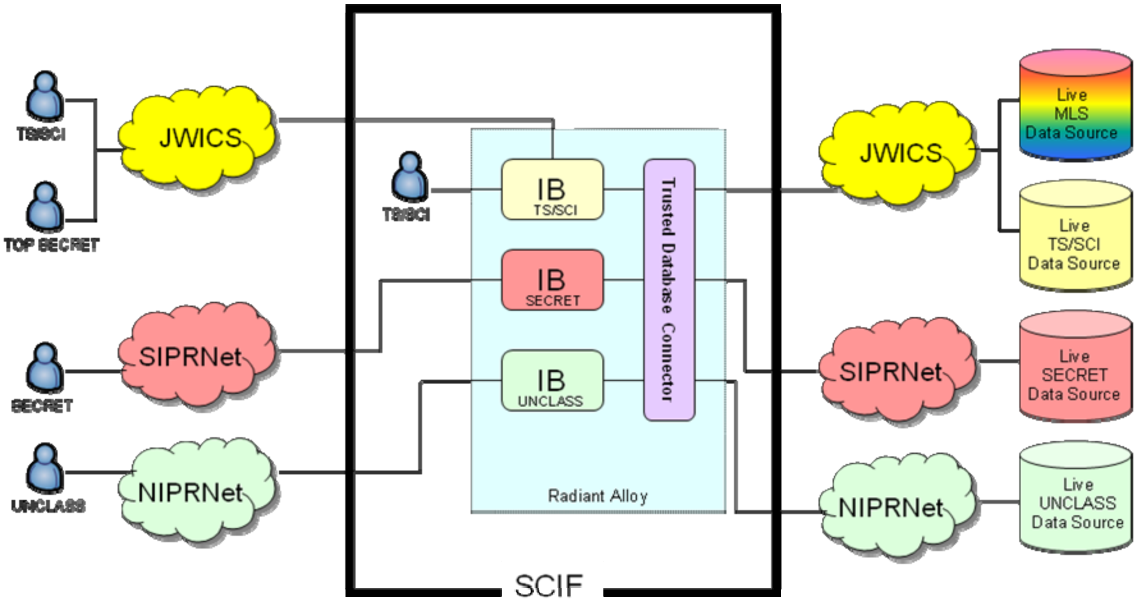


Figure 10. Radiant Alloy Architecture for High-Assurance, from [7]

Collection and dissemination of information are handled well by single-level systems, within homogenous domains. When we move out of that context, our information sharing abilities become more challenging and the sharing of information can rapidly degrade if our systems are not designed to accommodate this sharing. This is attributable to both our military's mindset toward not giving away anything the enemy might use against us, as well as the technological difficulty of creating an information system that we can trust,[17] to reliably and securely share data without compromise. Architectures, particularly distributed and service-oriented architectures, pose a challenge

[17] Trust is a measure of trustworthiness, relying on the evidence provided; where an entity is trustworthy if there is sufficient credible evidence leading one to believe that the system will meet a set of given requirements [3].

to meeting security requirements for a multilevel system. However, security SOA systems are experiencing greater integration with current MLS systems. This integration results in a greater need for solid, technical-based security assurances for the architecture. This is based on the additional security requirements that a MLS system is designed to meet, which provide some level of trust and assurance to the user. It is with this intent that we developed our Maritime Domain Awareness scenario.

IV. USAGE SCENARIOS

A. MARITIME DOMAIN AWARENESS HIGH LEVEL ALERT SCENARIO

Our application scenario comes from Maritime Domain Awareness (MDA)[18], which would encompass the proposed Radiant Alloy system. The SysML[19] view of this domain example is shown in Figure 11.

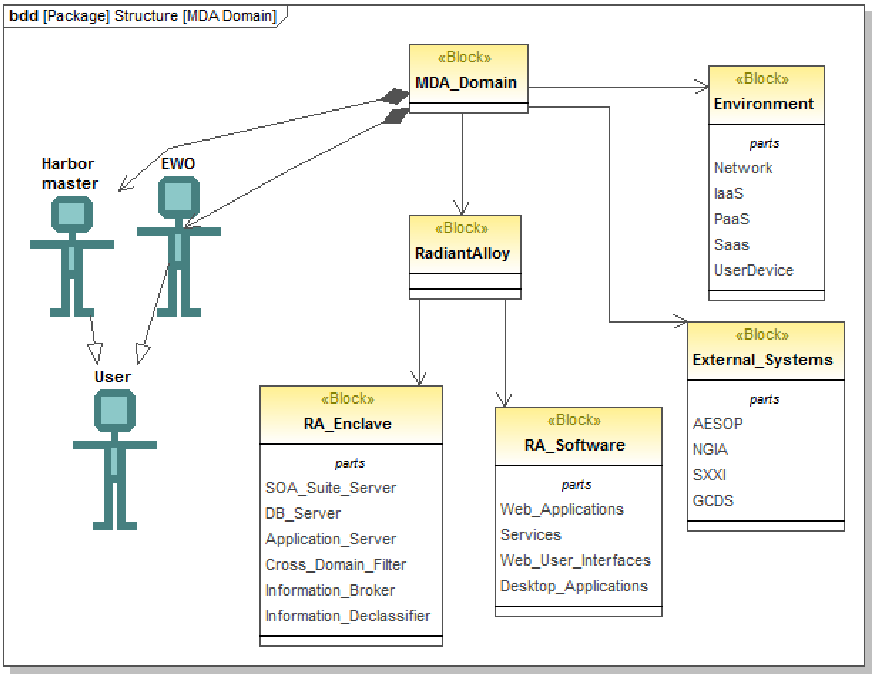


Figure 11. MDA Domain SysML Diagram

In the scenario, ships depart Southeast Asia and sail toward ports on the West Coast of the United States. Under normal circumstances the harbormasters at these ports share information about the ships expected departure and arrival times and contents of the cargo. In addition, other external resources develop information (which originates in

[18] Maritime Domain Awareness is the effective understanding of anything associated with the maritime domain that could impact the security, safety, economy, or environment of the United States.

[19] The Systems Modeling Language (SysML) is general purpose visual modeling language for systems engineering applications. SysML supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems.

different systems with higher classification levels that emits system-high alerts) that may mandate extensive searches of ships that take this normal course, the information ought to be shared in a way that hides the sources (and methods of collection) of the information, and make it appear to be information received through pre-arranged communication channels. For this research, assume that USS Antietam (CG-54), which is performing an interdiction mission in international waters in the Pacific Ocean, can generate high-level alerts about other ships. The Use Case (shown in Figure 12) can be extended to the application scenario as follows:

- The ship Globalstar7 becomes identified as a Vessel of Interest (VOI).
- The USS Antietam encounters Globalstar7 in international waters and stops the ship as part of its mission or merely observes the ship via other means.
- The Electronic Warfare Officer (EWO) onboard the USS Antietam, utilizing available sources of information, obtains data about the VOI and aggregates data from these sources.
- The EWO updates the track for the Globalstar7 in accordance with his duties and responsibilities.
- The EWO sends an Alert Notification to the destination port's Harbormaster.
- The notification advises the destination port (San Diego) to watch for the VOI (i.e., Globalstar7) and alert upon arrival.

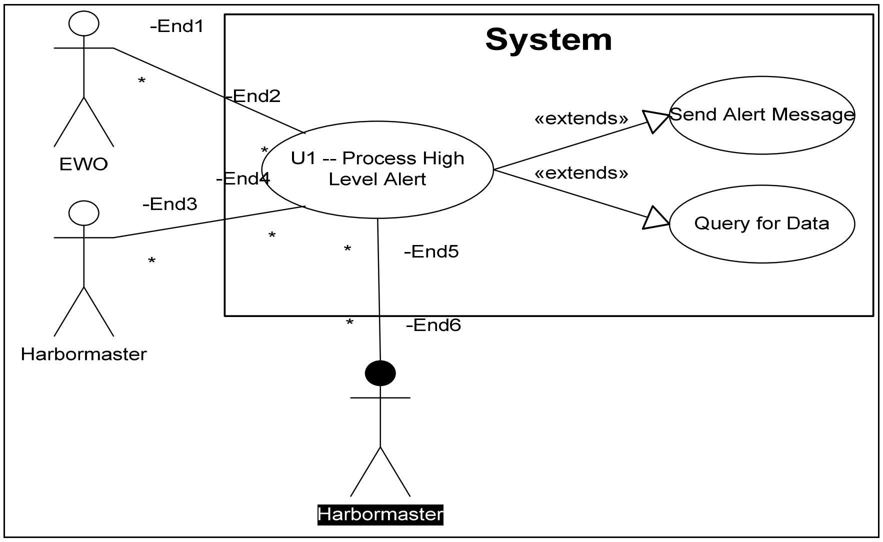


Figure 12. High Level Alert Use Case Diagram

Within the system boundaries an Information Broker is the key respondent to access requests, information queries, and data access. The system architecture for the MDA scenario and the Use Case is shown in Figure 13 and Figure 14.

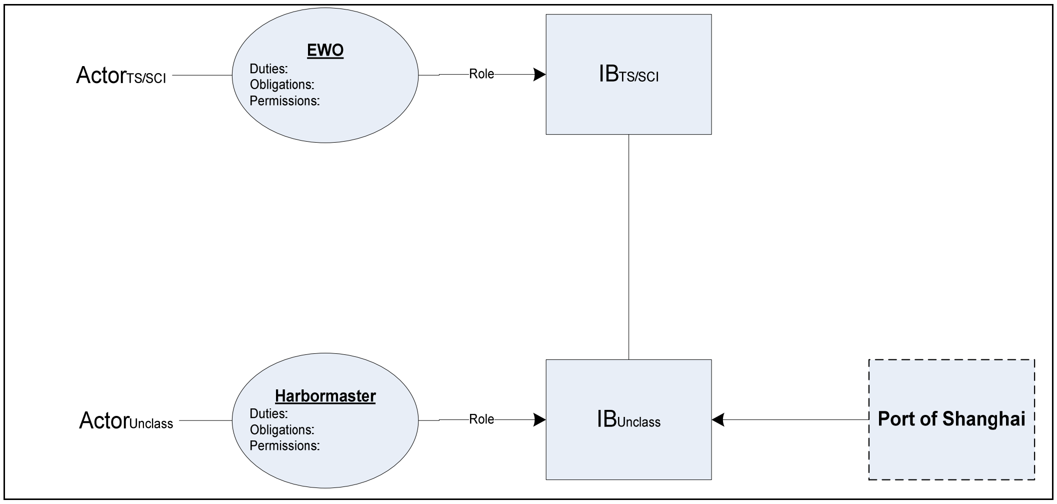


Figure 13. Graphical Representation of Use Case System Architecture

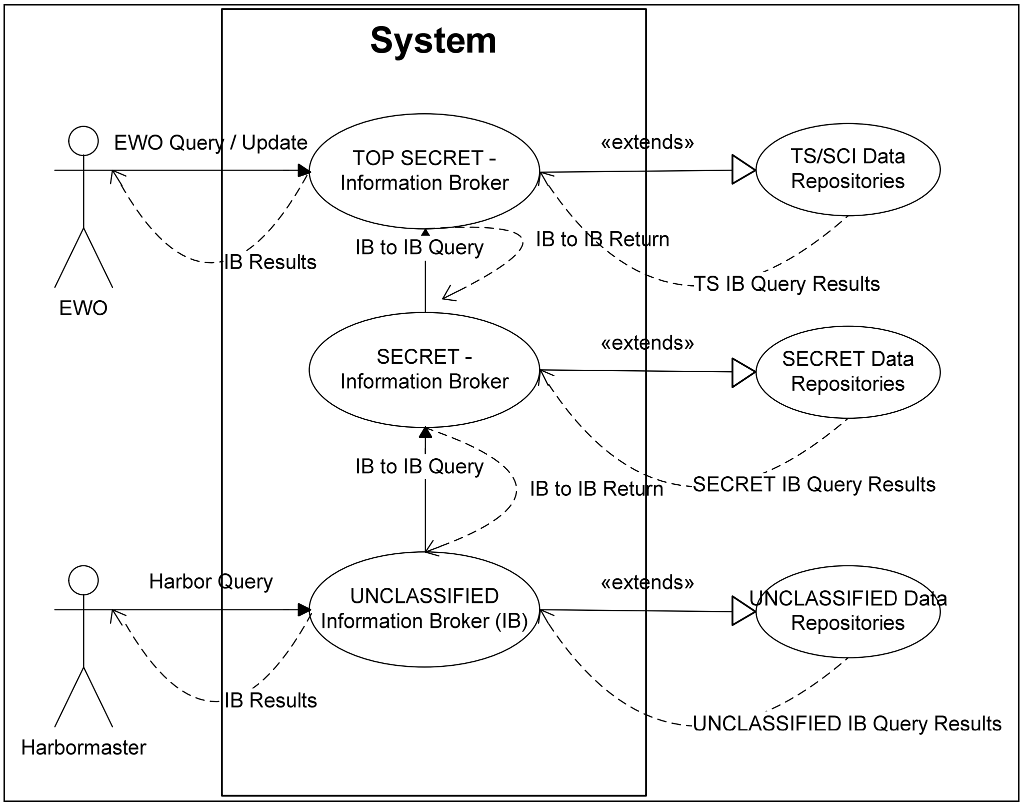


Figure 14. Use Case for IB System Architecture

While considering the system architecture and the overall scenario for the Use Case, we must illustrate the actions for all actors with a UML Sequence Diagram, as shown in Figure 15.

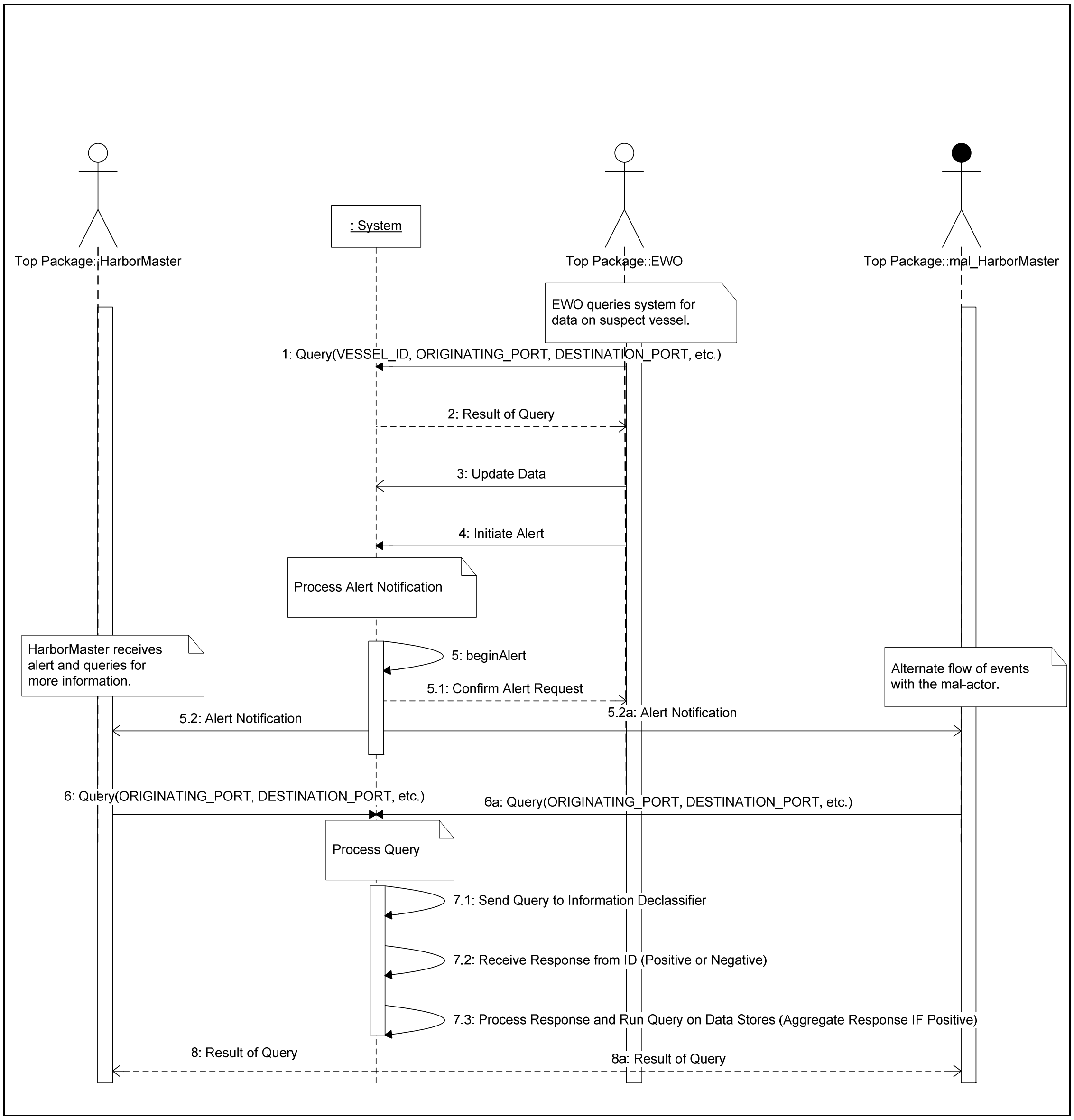

Top Package::HarborMaster
: System
Top Package::EWO
Top Package::mal_HarborMaster
EWO queries system for data on suspect vessel.
1: Query(VESSEL_ID, ORIGINATING_PORT, DESTINATION_PORT, etc.)
2: Result of Query
3: Update Data
4: Initiate Alert
Process Alert Notification
HarborMaster receives alert and queries for more information.
5: beginAlert
Alternate flow of events with the mal-actor.
5.1: Confirm Alert Request
5.2: Alert Notification
5.2a: Alert Notification
6: Query(ORIGINATING_PORT, DESTINATION_PORT, etc.)
6a: Query(ORIGINATING_PORT, DESTINATION_PORT, etc.)
Process Query
7.1: Send Query to Information Declassifier
7.2: Receive Response from ID (Positive or Negative)
7.3: Process Response and Run Query on Data Stores (Aggregate Response IF Positive)
8: Result of Query
8a: Result of Query

Figure 15. Sequence Diagram for High Level Alert MDA Scenario

The sequence diagram also depicts the mal-actor as an alternative flow of events, since he is not overtly trying to misuse the system. His actions are merely in the performance of his duties and show the sequencing that results from the system interaction.

1. High Level Alert Use Case

During the performance of duty on the USS Antietam, the EWO has (Top Secret level) data via a variety of intelligence means (e.g., electronic, signals, and human [ELINT, SIGINT, HUMINT]) to confirm ship, track and other details which from the composition of the data sources can give an aggregate picture of the VOI. These sources can also be used to compose other attributes, such as track history, port history, or even Measurement and Signature Intelligence (MASINT) data signatures that may be lacking from the original information about the ship. This combined data represented in a new and more detailed view, then categorizes at a higher classification level than originally intended (i.e., Top Secret instead of Unclassified). Table 5 shows the detailed Use Case for the Electronic Warfare Officer serving onboard the USS Antietam.

Table 5. High Level Alert EWO Use Case

Item	Contents
Use Case Name	U1–High Level Alert
Actors	EWO (Top Secret)
Brief description	EWO is aboard a U.S. Navy Cruiser class ship performing an interdiction mission. EWO is responsible for information and tracking of all ships in a theater of operation using ELINT, SIGINT, HUMINT, IR, and imagery. For vessels of interest, the EWO can query and update information in the system regarding those ships via his array of data collection sources. For particular vessels of interest, the EWO can put a watch on those ships at arriving ports and monitor via intelligence methods available to him.
Flow of events	1. EWO (Top Secret) logs in to the system (IB) 2. IB authenticates user to PDP 3. IB creates a session for the user 4. EWO requests data via a query to the IB 5. IB queries data stores 6. IB provides data (TOP SECRET and below) to the EWO 7. EWO synthesizes data and updates track and vessel information.

Item	Contents
	8. EWO requests IB to write data into the system. 9. IB verifies EWO to PDP 10. IB performs write operation to data store
Alternative flow of events	1. EWO (TOP SECRET) logs in to the system (IB) 2. IB authenticates user to PDP 3. IB creates a session for the user 4. EWO requests data via a query to the IB 5. IB queries data stores 6. IB provides data (TOP SECRET and below) to the EWO 7. EWO exits the system
Precondition	User is an authorized user of the system and has a valid role (EWO) established in the system. TOP SECRET level access to the system is available via a terminal.
Post-condition	User is authorized user of the system and has a valid role (EWO) established in the system.

The Unclassified level hosts another actor for the Use Case. This actor is the Harbormaster (role) for the port of San Diego where the Alert Notification will be directed. The Harbormaster is responsible for all inbound and outbound traffic for the entire port. He must manage berthing space and overall flow in the performance of his role. The Harbormaster is also obligated to comply with alert messages and adhere to his primary responsibilities. Based on an Alert Notification, the Harbormaster is instructed to watch for Globalstar7 arriving at an approximate date-time group (DTG)[20] from the Port of Shanghai, China and report upon arrival. Table 6 depicts the detailed Use Case actions for the Harbormaster.

[20] This is a messaging time format used by the U.S. military.

Table 6. High Level Alert Harbormaster Use Case

Item	Contents
Use Case Name	U1–High Level Alert
Actors	Harbormaster (Unclassified)
Brief description	Harbormaster is responsible for inspection and management of all ships entering and leaving the port. He must know: Existence of the vessel, scheduled arrival, reported size, port of origination, other? Using the harbormaster role; allows for queries to the system for information pertaining to origination port, destination port, expected arrival time (window), size/class of vessel, and name.
Flow of events	1. User (UC) logs in to the system (IB) 2. IB authenticates user to PDP 3. IB creates a session for the user 4. User requests data via a query to the IB 5. IB finds the relevant data store a. Managed by this IB b. Managed by another IB 6. IB queries the data store(s) 7. IB provides data that meets the query criteria and that is authorized per the Role-Permission mapping and security level for the Harbormaster
Alternative flow of events	1. User (UC) logs in to the system (IB) 2. IB authenticates user to PDP 3. IB creates a session for the user 4. User requests data via a query to the IB 5. IB finds the relevant data store 6. Managed by this IB 7. Managed by another IB 8. IB queries the data store(s) 9. Harbormaster is not authorized any data and IB provides a negative data found response
Precondition	User is an authorized user of the system and has a valid role (Harbormaster) established in the system. Access is available via a terminal to the information system
Post-condition	User is authorized user of the system and has a valid role (Harbormaster) established in the system. Data is used to perform the duties of Harbormaster

These two actors in these two Use Cases (i.e., EWO and Harbormaster) that are operating at different security levels combine their roles to perform a standard maritime mission. These actions, however, and the actors' associated duties and responsibilities, also establish a basis for our Misuse Case.

2. High Level Alert Misuse Case

The Misuse case is designed to reveal a security problem, where the system is not performing its intended use. The mal-actor in this Misuse Case is actually one of our actors from the use case, the Harbormaster. After receiving an Alert Notification concerning the vessel of interest, the Port of San Diego Harbormaster queries his system for all ships destined to his port and originating from the Port of Shanghai. But, the Globalstar7 does not show on this list–from this the Harbormaster can infer that someone knew about this ship and its questionable nature (since he was told to watch for its arrival) but he was not authorized this information from his access to the system. Now, there is an unintended flow (existence) of Top Secret information to the lower (Unclassified) level, and the creation of a covert channel, violating the *-property (i.e., *no write down*) of the BLP model. The Misuse Case system architecture overview is shown in Figure 16 and Figure 17, while the detailed Misuse Case is shown in Table 7.

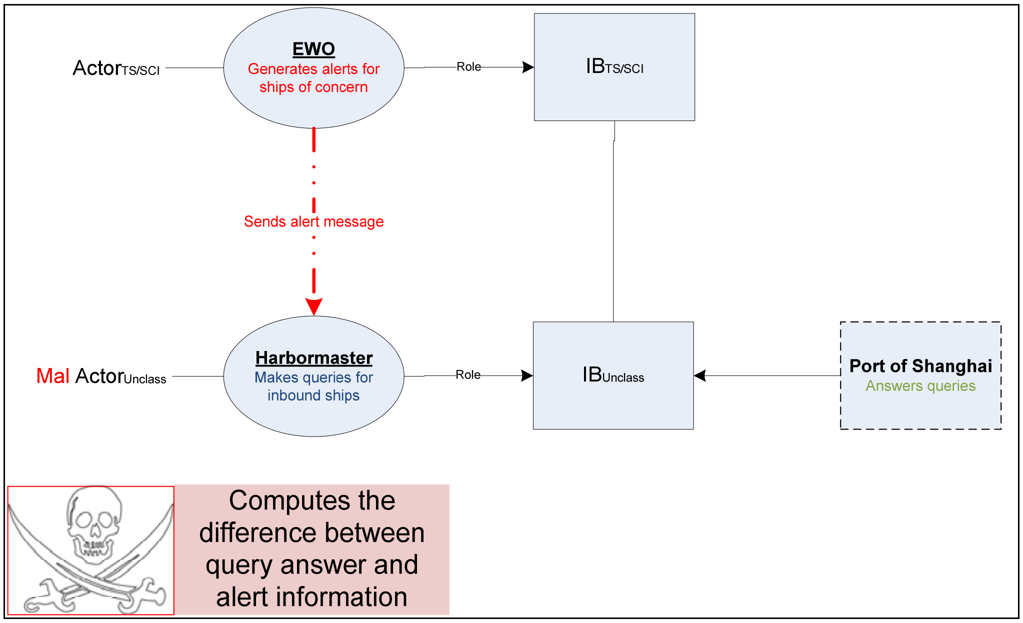


Figure 16. Graphical Representation of Misuse Case System Architecture

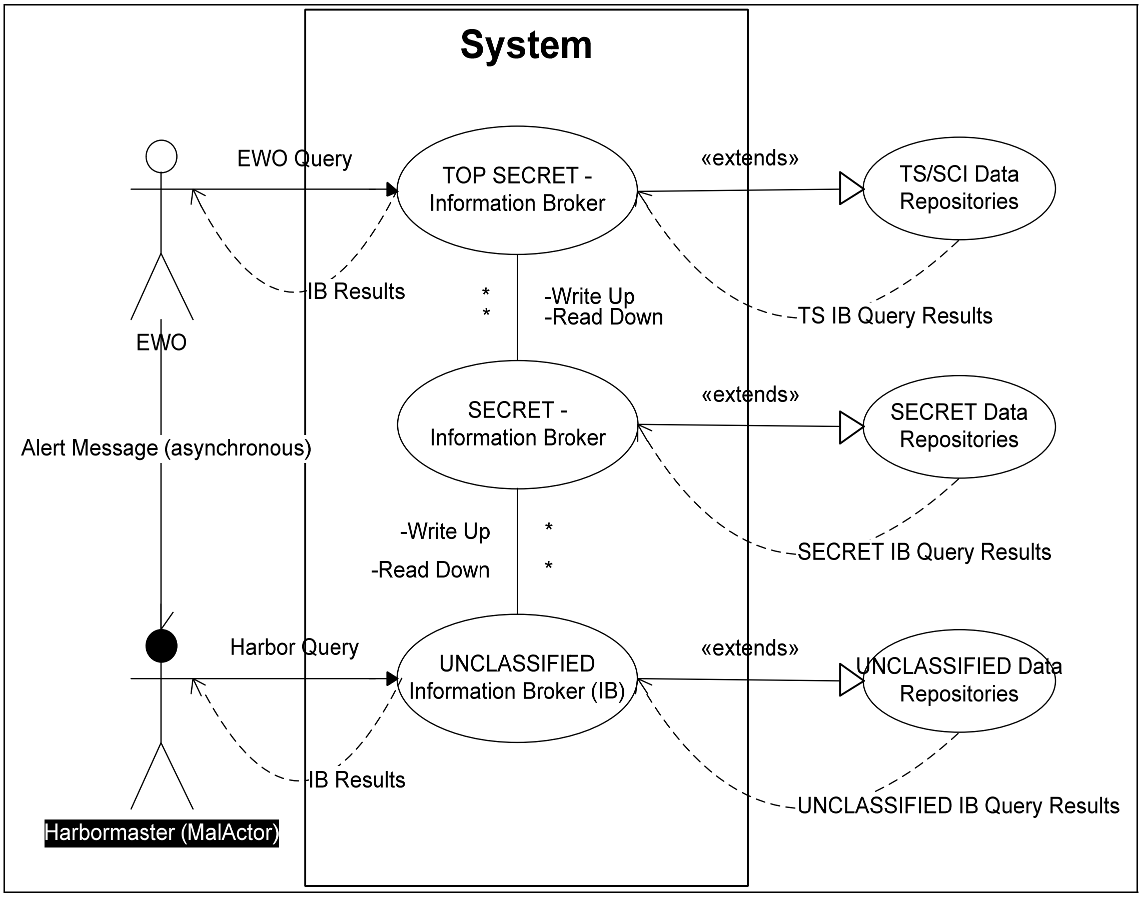


Figure 17. Misuse Case System Architecture

Table 7. High Level Alert Misuse Case

Item	Contents
Misuse Case Name	MU1–High Level Alert (Transition to Lower Level)
Actors	Harbormaster (Unclassified), Electronic Warfare Officer (TOP SECRET)
Brief Description	The Harbormaster is informed of a Watch Alert for a particular ship (Globalstar7), inbound from Shanghai, China to arrive at an approximate DTG. Harbormaster logs in to the system and requests data from the system (using one of his authorized queries) for all ships arriving to his port (San Diego) from the Port of Shanghai, China. The system (IB) provides data to the Harbormaster based upon his role and clearance level, but that ship (Globalstar7) is not in the data set provided by the system. Around the approximated DTG the Globalstar7 ship arrives in his port. The Harbormaster can infer that he was not privileged to this information and that the system has other types of collection / reporting assets that created and verified this ship's track.
Flow of Events	1. User (UC) logs in to the system (IB) 2. IB authenticates user to PDP 3. IB creates a session for the user 4. User requests data via a query to the IB 5. IB finds the relevant data store a. Managed by this IB b. Managed by another IB 6. IB queries the data store(s) 7. IB provides data that meets the query criteria and that is authorized per the Role-Permission mapping and security level for the Harbormaster
Alternative Flow of Events	1. User (UC) logs in to the system (IB) 2. IB authenticates user to PDP 3. IB creates a session for the user 4. User requests data via a query to the IB 5. IB finds the relevant data store a. Managed by this IB b. Managed by another IB 6. IB queries the data store(s) 7. Harbormaster is not authorized any data and IB provides a negative data found response

Item	Contents
Precondition	User is cleared (Unclassified) and has a role (Harbormaster) in the system
Assumption	A ship exists that is of a particular concern and has a track related update performed by a higher clearance level user.
Exploited Vulnerability	An out of band signaling channel exists and is utilized in the performance of duties.
Worst Case Threat	The Harbormaster can make an inference from the out of band signaling about the system and divulge privileged information. Could also divulge the identity of the sender (for the Alert)
Capture Guarantee	No out of band (outside of system) communication is allowed (via phone, email, etc.). All communications must go through the system (and the IB).
Related Business Rules	Scope of duty for the EWO to report the threat as an Alert notification to the harbor and within the scope of duty for the Harbormaster to acknowledge and respond/watch for the vessel indicated in the alert notification.
Potential Misuse Profile	This misuse could happen accidentally and does not require ability or skill beyond that of a normal Harbormaster user who can query the system and possesses an average intellect to infer that someone knew about the ship's arrival, but it was not in the system. Intentionally, the misuse can be exploited using other authorized queries (per the Harbormaster role) against other facts (i.e., DTG of expected arrival, size of ship, destination port, or ship name).
Stakeholders and Threat	System Designers / Developers–system implementation is compromised because of the out of band signaling used Certifiers & Accreditors–system does not provide assurance and meet requirements to pass certification Trainers for the users (Actors) in the system–users do not perform their duties as instructed and open the out of band signaling channel to compromise the system
Scope	Maritime scenario

This scenario is certainly not limited to this single misuse. It is merely being offered as an example scenario to drive the Use Case analysis and to provide a basis for the system's response via the Security Use Case.

3. High Level Alert Security Use Case

Our problem that originates from this Misuse Case is the uncontrolled observability of differing security domains. This violates the safety of both access control models within the system and the specified information-flow control policies. The combination of use and misuse necessitates a Security Use Case to *mitigate*, *detect*, or *prevent* an associated misuse [64, 65]. This Security Use Case represents the system's method of dealing with the vulnerabilities the Misuse Case exploited within the system. By analyzing the interactions of Use and Misuse, and then the associated Security Use Case–we can develop better system requirements that will prevent security breaches and provide some type of safety guarantee for our information. In this instance, the Misuse Case we presented exposes that we have an information flow and more importantly an information leakage problem within our system. The detection and mitigation of this leakage requires a change to the system architecture to account for the Security Use Case's handling of the Use and Misuse Cases. The primary means to eliminate this misuse is via the rerouting of information in the system, as well as adding a new architectural element; the Information Declassifier. The detection of queries and alerts within the system must be supported by changing the architecture. The revised system architecture for the Security Use Case is shown in Figure 18 and Figure 19. For all system data queries, the Information Broker must provide filtered information contained in higher level IB alerts (or parts of information from those alerts that need to be shared with the levels below) that need to be read by the lower levels, and inject them into the lower level data query so that the covert channels created by the missing-information will not be there. The process can be done in two stages:

1. The views and queries available to the lower levels (i.e., Secret level) will be made available to the Top Secret level at design time.
2. Create information downgrading rules between each (High, Low) level pair.

This allows the information that the lower classification level user is authorized to view to still be viewable (through the Alert and use of the Information Declassifier), instead of suspiciously absent because a higher classification level user "touched" the data and now they cannot Read it (even though it is within the scope and permissions of their Role's duties). This provides a user with inference ability through the lack of information resultant from an authorized query to the system.

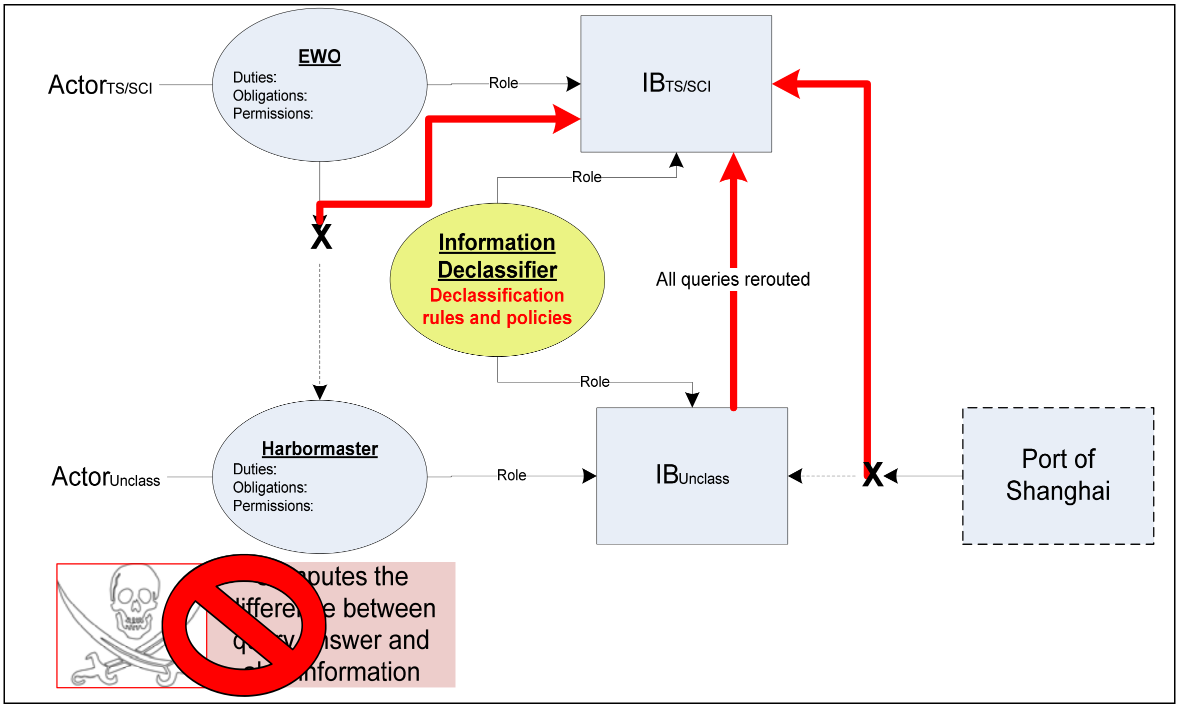


Figure 18. Graphical Representation of Security Use Case System Architecture

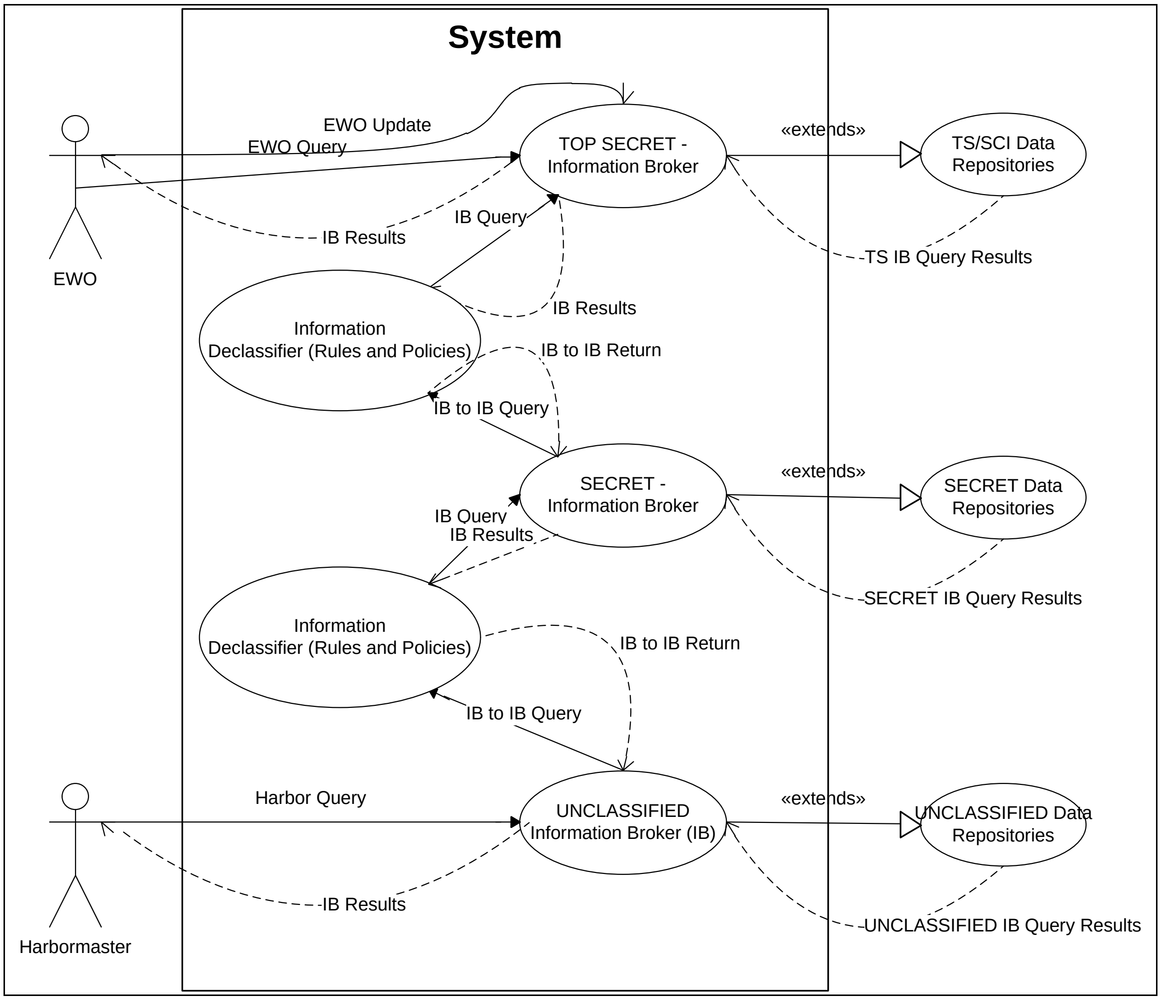


Figure 19. Security Use Case for IB System Architecture

The removal of higher clearance level attributes is essential to maintaining the safety of the system and hinges on the lower level IB checking with a higher level IB. Each IB must successively iterate the query upward until the highest level IB is reached. This is, most likely, the global instance of the Top Secret IB. It is this IB that will give the authoritative answer when lesser IBs ask how to respond to queries. This IB will have to inject the presence of the Globalstar7 from the Port of Shanghai, in order to prevent this inference disclosure. This information injection will originate from the original Alert that is processed by the system during the execution of the EWO's duties and obligations to report on specified criteria. But, it cannot just use the entire alert message and must filter the contents of the message based on role permissions for the user who executed the initial query that have been mapped to the user's role and validated with the PDP. The detailed Security Use Case is presented in Table 8, Table 9, Table 10, and Table 11.

Table 8. High Level Alert Security Use Case (Alert Detection)

Item	**Contents**
Use Case Name	SU1–High Level Alert Security Use Case
Actors	EWO, Harbormaster
Brief description	Three sub-cases exist for this Security Use Case where the system must <*detect*> interactions at different touch points. These sub-cases include: 1. The system detects an Alert message is being posted from a user (at any level). 2. The system detects a user query submitted to the Information Broker (at any level). Two sub-cases exist: positive presence of data, and negative presence of data. 3. The system detects an Inter-IB query (a lower level IB queries to a higher level IB). Two cases exist: positive presence of alert messages pertaining to the query, and negative presence of alert messaging pertaining to the query.
Flow of events	1. The system <*detects*> an Alert message and begins processing IAW SU1.1
Alternative flow of events	1. The system <*detects*> a query and begins processing IAW SU1.2 (User-IB) or SU1.3 (Inter-IB)
Precondition	Users of the use and misuse cases are authorized users of the system and have valid roles (EWO and Harbormaster) established in the system with a common subset of permissions mapped from the roles. Access to the system is available to all users.
Post-condition	User is authorized user of the system and has a valid role established in the system.

Table 9. High Level Alert Security Use Case (Alert Detection)

Item	Contents
Use Case Name	SU1.1–High Level Alert Security Use Case (Alert Detection)
Actors	EWO, Harbormaster
Brief description	The system detects an Alert message is being posted and (a) examines the message for its distribution (b) maps permissions from the distribution roles (those who it is intended for) for accesses to those fields that are contained in the message thru multiple queries to the PDP (one query for each role / user on the distribution list) (c) removes parts of the message (data fields) that are not permitted for the roles permission mapping (d) transmits the message to the roles listed in the distribution.
Flow of events	1. An Alert message is posted to the system and stored at the current user's classification level. 2. The distribution list (recipients) of the message is examined. 3. Each recipient is checked (by role) with the PDP for permissions on that message by field. 4. Individual fields and data are filtered from the message based on the returned permission set (this could be an empty set). 5. Message is delivered to intended recipients with unauthorized content filtered.
Alternative flow of events	1. An Alert message is posted to the system 2. The distribution list (recipients) of the message is examined. 3. Each recipient is checked (by role) with the PDP for permissions on that message by field. 4. Individual fields and data are filtered from the message based on the returned permission set (this could be an empty set). 5. Message contains no data available for delivery to a user and is stored at the current classification level.
Precondition	Users of the use and misuse cases are authorized users of the system and have valid roles (EWO and Harbormaster) established in the system with a common subset of permissions mapped from the roles. Access to the system is available to all users.
Post-condition	User is authorized user of the system and has a valid role established in the system.

Table 10. High Level Alert Security Use Case (User-IB Query Detection)

Item	Contents
Use Case Name	SU1.2–High Level Alert Security Use Case (User-IB Query Detection)
Actors	EWO, Harbormaster
Brief description	There are two alternative flows for this Use Case. The system detects a query has been submitted by a user to the Information Broker and (a) checks for validity of the query based on permissions mapped to the user role with a permission query to the PDP (b) checks with the higher level Information Broker for Alert messages that pertain to fields or values in the query that were validated by the PDP permission response (c) removes (filters) information from the query that is not permitted for viewing by the role of the requestor (d) provides the remaining information from the alert to the lower level IB (e) queries its own current level repositories for additional data matching the query and allowable for the role based permissions (f) aggregates the items from (d) and (e) to provide data to the user that is allowable via the role permissions assigned to the user. Alternatively, the system detects a query has been submitted by a user to the Information Broker and (a) checks for validity of the query based on permissions mapped to the user role with a permission query to the PDP (b) checks with the higher level Information Broker for Alert messages that pertain to fields or values in the query that were validated by the PDP permission response (c) negative response for alert message presence to the lower level IB (d) queries its own current level repositories for additional data matching the query and allowable for the role based permissions (e) provides data matching the query and allowable via the role permissions assigned to the user.
Flow of events	1. The system detects a query has been submitted by a user to the Information Broker 2. Checks for validity of the query based on permissions mapped to the user role with a permission query to the PDP 3. Generates an Inter-IB query, which checks with the higher level Information Broker for Alert messages that pertain to fields or values in the

Item	Contents
	query that were validated by the PDP permission response 4. Removes (filters) information from the query that is not permitted for viewing by the role of the requestor 5. Queries its own current level repositories for additional data matching the query and allowable for the role based permissions 6. Aggregates the Inter-IB query response with the original query response at the current IB level to provide data to the user that is allowable via the role permissions assigned to the user.
Alternative flow of events	1. The system detects a query has been submitted by a user to the Information Broker 2. Checks for validity of the query based on permissions mapped to the user role with a permission query to the PDP 3. Generates an Inter-IB query that checks with the higher level Information Broker for Alert messages that pertain to fields or values in the query that were validated by the PDP permission response 4. Receives a negative response for alert message presence from the Inter-IB query 5. Runs the original query against its own current level repositories for data matching the query and allowable for the role based permissions Provides data matching the query and allowable via the role permissions assigned to the user or provides a negative response to the user for the data query.
Precondition	Users of the use and misuse cases are authorized users of the system and have valid roles (EWO and Harbormaster) established in the system with a common subset of permissions mapped from the roles. Access to the system is available to all users.
Post-condition	User is authorized user of the system and has a valid role established in the system.

Table 11. High Level Alert Security Use Case (Inter-IB Query Detection)

Item	Contents
Use Case Name	SU1.3–High Level Alert Security Use Case (Inter-IB Query Detection)
Actors	EWO, Harbormaster
Brief description	The system detects an query from a lower level IB to a higher level for alert message data (which includes role and permission information) and (a) submits a similar type query to its higher level IB (b1) aggregates a positive response with any additional alert messages that exist at its own level or (b2) after a negative response from a higher level the IB queries its own level for alert message information pertaining to the query and allowable via the role permissions (c) provides either a negative response to the lower level IB or the resultant data from (b1) or (b2) to the lower level IB.
Flow of events	1. The system detects an Inter-IB query from a lower level IB to a higher level for alert message data (which includes role and permission information) 2. Submits a similar type query to its higher level IB (unless it is the authoritative IB) 3. Aggregates a positive response to the Inter-IB query with any additional alert messages that exist at its own level 4. Filters the information not permissible for the user's role permission set. 5. Provides the resultant data set from the query to the lower level IB.

Item	Contents
Alternative flow of events	1. The system detects an Inter-IB query from a lower level IB to a higher level for alert message data (which includes role and permission information) 2. Submits a similar type query to its higher level IB (unless it is the authoritative IB) 3. Receives a negative response from a higher level and then queries its current level for alert message information pertaining to the query and allowable via the role permissions 4. Filters the information not permissible for the user's role permission set. 5. Provides either a negative response to the lower level IB if the permission to data mapping is empty. 6. Provides a filtered result of the data set to the lower level IB.
Precondition	Users of the use and misuse cases are authorized users of the system and have valid roles (EWO and Harbormaster) established in the system with a common subset of permissions mapped from the roles. Access to the system is available to all users.
Post-condition	User is authorized user of the system and has a valid role established in the system.

4. Impact of Misuse and Mitigation

Overall, the Misuse Case exposes an information flow and more importantly an information leakage problem within our system. The leakage is an inference about cross-domain information. The detection and mitigation of this leakage requires a change to the system architecture to account for the Security Use Case's handling of the Use and Misuse Cases. The primary means to eliminate this misuse is via the rerouting of information in the system, as well as adding a new architectural element; the Information Declassifier. The ID must act as an intermediary between varying security domains, represented by an Information Broker (IB), to process queries from lower classification levels and to pull data pertaining to Alert Notifications from higher classification level data stores. This is necessary, because, if a higher level user *touches* the data pertaining to a VOI, the lower level user can no longer *view* that data, that they are entitled to (in performance of their duties to fulfill a Role). In order to prevent such security violations, Radiant Alloy needs to use an information declassifier. In this research, we provide a policy-based framework to do so using RuleML.

This revised system architecture is a proposed goal of this research and is shown in Figure 20. It is in this re-architecting that we have rerouted all system queries from both users and existing domain level information brokers, in order to prevent the misuse of our system. Although this figure depicts a singular Information Declassifier (ID), we must provide for ID replication throughout the system and particularly between each information domain.

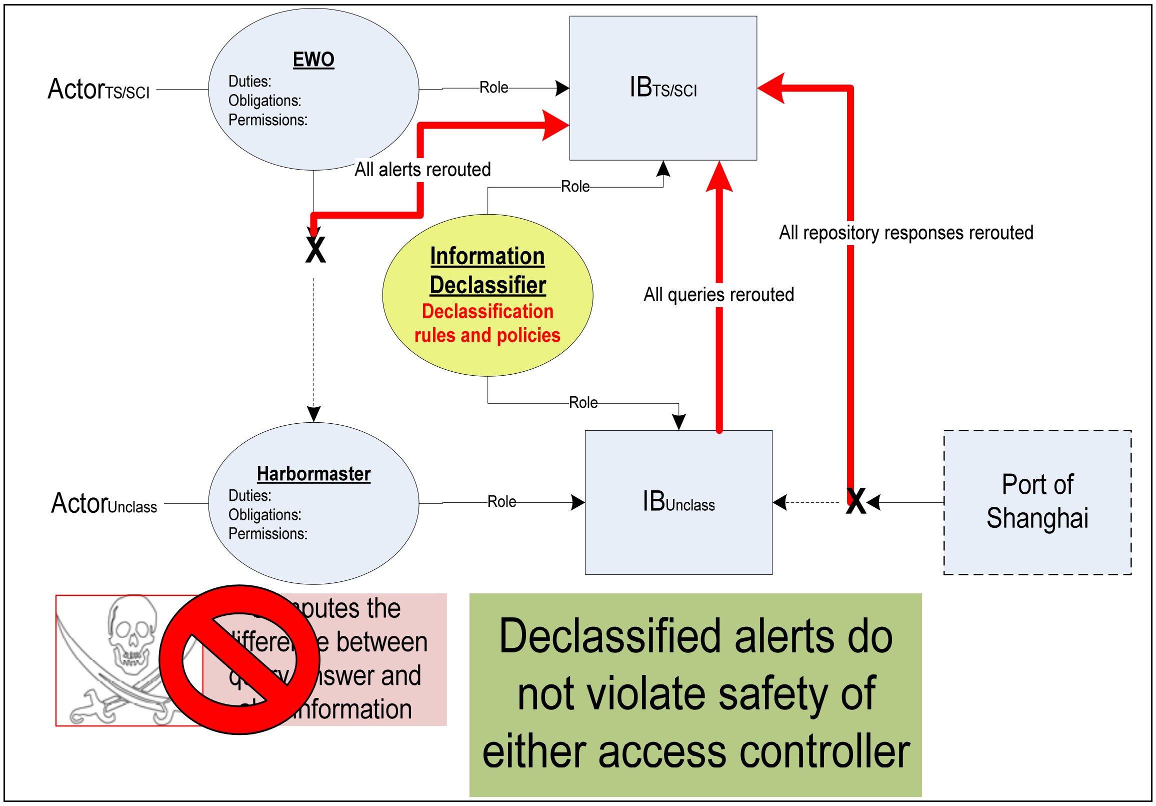


Figure 20. Graphical Representation of System Architecture for Leakage Mitigation

Because of this inter-domain requirement, the ID serves as a proxy for all messaging within the system. It is here where the ID will take cleared messages and include them in referenced queries, to which the lower level IB itself should not see outright (because of BLP's Simple Security Property). This use of an Information Declassifier adds complexity to our system design and requires additional interactions between domain level information brokers. Between each domain we must have an ID residing to request and deliver information from higher domains. This will be iterative from the lowest IB domain level (e.g., Unclassified) up to the highest level IB domain (e.g., Top Secret), where an ID will be able to query for Alert notifications (or other data) and be able to provide that as required to a lower level IB domain. A state chart representation of the query and alert process with the interactions between separate IBs and an ID is shown in Figure 21. The start state is denoted by the solid black circle, the arrows indicate transition flow, and the double circles, which encompass the IBs and ID denote accepting states within the statechart.

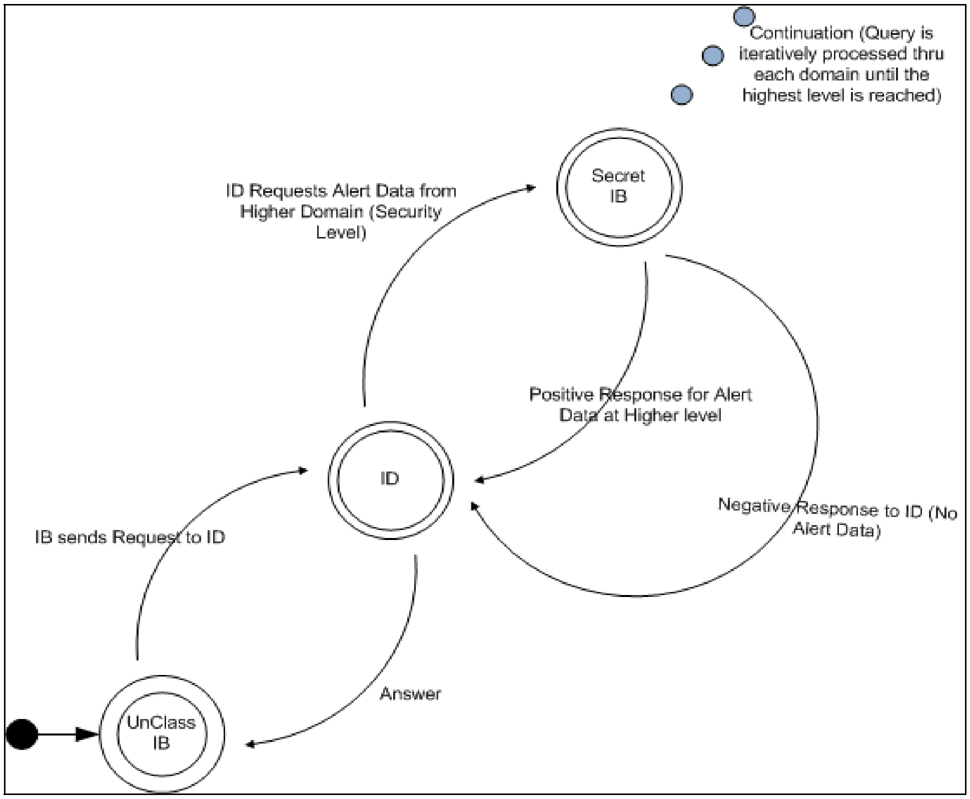

Continuation (Query is iteratively processed thru each domain until the highest level is reached)
Secret IB
ID Requests Alert Data from Higher Domain (Security Level)
Positive Response for Alert Data at Higher level
ID
Negative Response to ID (No Alert Data)
IB sends Request to ID
Answer
UnClass IB

Figure 21. Statechart for Information Broker and Information Declassifier Process

As each query is received at an individual domain’s IB, the individual IB will first perform its duties to authenticate the user, and the user’s permissions for the requested data, based on both the user’s role permission set and the classification level of the information. After a query is acknowledged as valid within the IB, the IB will forward the user query to the ID, where it will then check with the higher level IB for information pertaining to that user query. The request to the higher level domain IB does not query the higher level data stores for new information, but merely checks for Alert messages within the system that pertain to a particular role or other mapping which we can establish through the rules that are expressed in the Information Declassifier. Each IB, upon receiving an ID request, will check with a higher level ID until the highest domain level of the system is reached. At each domain level upon an ID query, the information broker will check its Alert messaging stores and provide either a positive response with the information or a negative response with no data. This will aggregate downward until the originating IB is reached. It is then that the IB will inject any alert response data with its own query response from domain level data fulfilling the user query. Information markings are not depicted in this flow, but the Information Broker is responsible for adding appropriate markings (at each security level) for any data that does not already contain markings. This will be appended to the data before the IB will return a response to a user

In this proposed architecture, we have no violation of BLP–since we have no write down, nor do we violate the RBAC policy within our classification domains. There is no indirect means of obtaining information either, since RBAC ensures that we are not seeing anything that the Role does not allow for in the first place (nothing that the actor is not already entitled to). The Harbormaster is entitled to examine information about ships that are entering his port. This is merely a cause of not being able to see information because a higher classification level entity last touched the data (where it cannot be written down *-Property) and he cannot read up to see it (Simple Security Property). As long as the system architecture is adhered to, the Information Declassifier will bridge the gap between classification levels without creating covert channels.

Making these changes to the system architecture requires several elements: detailed specification of the Information Declassifier and its actions (using RuleML) via domain-specific declassification policies and rules, representation of query translations, representation of system alert notifications, a revised definition of safety for our new architecture, and establishment of test cases that ensure the functionality of the Use Cases and prevent the Misuse Cases within the re-architected system. From this we can regain the Safety of our information flows within the system and prevent information leakage between domains for all future system that adhere to our re-architecting efforts.

5. Conclusion

This Use-Case analysis and its application demonstrate the criticality of architecting an information broker that is integrated with the security requirements for high-assurance, while also maintaining the core functionality of the system without degradation. In general, we design a system to meet specific requirements as they are provided by a scenario that we wish to support. It is from these requirements that we generate a desired policy that we want to enact within the system, and as a result, we select an access control or security model that most closely approximates the intended policy. In the policy that is created, we expect to achieve a safety of information flows by using an established security model. By mapping the policy to the model and then extending our safety analysis from the underlying model to the overarching security policy, we can provide a safety guarantee for our system. It is with this safety guarantee in mind, that the designers select an architecture and begin the implementation of the associated security policy.

By integrating the security requirements when deciding on an access control model, and, in this case, using a mixed access control model, while developing a rule set for integration to our security policy to prevent the described Misuse cases. The execution of the Security Use Case will originate with the Information Declassifier and the rules associated with its use as they are mapped to the Use and Misuse Cases for the desired system behaviors to maintain the safety property.

V. QUERY AND ALERT MESSAGING

A. XML DATA SET

In Radiant Alloy, users are able to enter queries to the Information Broker that correspond to information or data required for their role. A representation of that data found in Radiant Alloy, can be expressed using eXtensible Markup Language (XML). In order to examine how this query ability could be influenced with the use of an ID, we have created a sample subset of XML data for vessel information and alert messages supporting our Use-Case scenario, for demonstrating the ability of the Use-Case analysis and ID rule generation to be extended to other scenarios.

Within the U.S. government, information markings are Defense Message System (DMS) General Service (GENSER) message classifications, categories and markings [66]. These have been added to the IB mechanism, which will actually perform the formal access control and apply classification, category, and dissemination control markings to data as it is pulled from repositories as appropriate. In this sample, data the marking field has been added and populated to provide the ability to parse queries appropriate for varying domains without the actual implementation of a multi-domain, replicated IB. The proper implementation of these marking requirements is to provide interoperability with respect to security classifications and categories of DMS GENSER messaging. Information regarding the sensitivity level and markings associated with message content needs to be conveyed to the user requesting information from the system. Additionally, the markings are required so that access control decisions can be made.

There are generally two methods for storing XML data in a repository. The first case is to utilize individual files for each element being described (i.e., a separate file for each vessel). While the second option is to establish a larger, single file with a parent element and list child elements inside (i.e., <VESSELS> as the parent element containing a <VESSEL> tag for each ship). The multiple file instance is more complex with respect to storage and querying, and the single storage instance is merely a degenerate case. In this example, we used the single document instance for both the Alerts and Vessels, as the method of storage does not impact the ability to exercise the rules of the ID.

1. Vessel Information

The vessel information used for this sample system is based on the Automatic Identification System (AIS). AIS provides the means for ships to electronically exchange data with other nearby ships and Vessel Traffic Services stations. AIS was developed to assist the vessel’s watch officers with a standardized information schema and allow maritime authorities to track ships. The AIS uses a vessel’s on-board navigational instruments to provide the data. This system reliably transmits information about vessels and ship tracking at fixed intervals of time. Certain information, such as the Maritime Mobile Service Identity (MMSI), Navigation Status, Speed, Latitude, Longitude, Heading, and Time Stamp, are transmitted every 2–10 seconds (depending upon speed) while underway and every 3 minutes while at anchor. Additional and less variable information is transmitted every 6 minutes, to include the IMO Number, Call Sign, Length, Width, Destination, and Estimated Arrival Time. The XML data code used in the Vessel.xml is shown in Listing 1.

```
<?xml version=“1.0” encoding=“UTF-8” standalone=“yes” ?>

<VESSELS>

<TITLE>Vessel Report</TITLE>
<MESSAGE>
<![CDATA[
      Listing of vessels resulting from your query.
]]>
</MESSAGE>

<!-- Sample Vessel used in Use Case: -->
<VESSEL>
<!-- Sample tracking data for vessels
-->
      <NAME_TXT>Name: </NAME_TXT>
      <NAME>Globalstar7</NAME>
          <MMSI_TXT>MMSI: </MMSI_TXT>
      <REGISTRY>China</REGISTRY>
      <MILITARY>No</MILITARY>
      <ORIGINATING_PORT>Shanghai</ORIGINATING_PORT>
      <TYPE>Commercial Vessel</TYPE>
      <SUB_TYPE>Container Ship</SUB_TYPE>
      <DEPARTURE>8/28/08 08:45</DEPARTURE>  <!-- UTC / month/date/year
hour:minute -->

<!-- AIS sends the following data every 2–10 seconds while underway
       depending on speed, and every 3 minutes while at anchor
-->
      <MMSI>412159177</MMSI>  <!-- Maritime Mobile Service Identity
412, 413 China-->
      <NAV_STATUS>Under Way Using Engines</NAV_STATUS> <!-- At Anchor,
Under Way, etc. -->
      <RATE_OF_TURN>0</RATE_OF_TURN> <!-- Right or Left, 0–720 degrees
per minute -->
      <SPEED_OVER_GROUND>10.2 Knots</SPEED_OVER_GROUND> <!-- 0 to 102
Knots, 0.1 increments -->
      <POSITION_ACCURACY>3 Meters</POSITION_ACCURACY>
      <LATITUDE>27.147145</LATITUDE>
      <LONGITUDE>-128.342285</LONGITUDE>
      <COURSE_OVER_GROUND></COURSE_OVER_GROUND> <!-- relative to true N
to 0.1 degree -->
          <TRUE_HEADING>27</TRUE_HEADING>    <!-- 0 to 359 degrees from
gyro compass  -->
      <TIME_STAMP></TIME_STAMP>    <!-- UTC time stamp for data -->

<!-- AIS broadcasts the following data every 6 minutes
-->
        <IMO_NUMBER>IMO 1234567</IMO_NUMBER>  <!-- # unchanged upon
transfer of registry -->
        <RADIO_CALL_SIGN>G7CS1</RADIO_CALL_SIGN>  <!-- up to seven
characters -->
        <TYPE_POSITIONING>LORAN-C</TYPE_POSITIONING> <!-- GPS, DGPS,
LORAN-C -->
    <LENGTH>200</LENGTH>
        <WIDTH>25</WIDTH>
    <DRAUGHT>25</DRAUGHT>
    <DESTINATION_PORT>San Diego</DESTINATION_PORT>  <!-- Max 20
characters -->
    <EST_ARRIVAL>10/1/08 13:00</EST_ARRIVAL>  <!-- UTC /
month/date/year hour:minute -->
    <CLASSIFICATION>Top Secret</CLASSIFICATION>
    <ALERT_PRESENT>Yes</ALERT_PRESENT>

</VESSEL>  <!--  -->

<!-- Sample Vessel used in Use Case: -->
<VESSEL>
<!-- Sample tracking data for vessels
-->
      <NAME_TXT>Name: </NAME_TXT>
      <NAME>USS ANTIETAM</NAME>
          <MMSI_TXT>MMSI: </MMSI_TXT>
      <REGISTRY>USA</REGISTRY>
      <MILITARY>YES</MILITARY>
      <ORIGINATING_PORT>San Diego</ORIGINATING_PORT>
      <TYPE>Military Vessel</TYPE>
      <SUB_TYPE>CG-54</SUB_TYPE>
      <DEPARTURE>0/0/00 00:00</DEPARTURE>  <!-- UTC / month/date/year
hour:minute -->

<!-- AIS sends the following data every 2–10 seconds while underway
       depending on speed, and every 3 minutes while at anchor
-->
      <MMSI>36677900</MMSI>  <!-- Maritime Mobile Service Identity 366
U.S.-->
      <NAV_STATUS>Under Way Using Engines</NAV_STATUS> <!-- At Anchor,
Under Way, etc. -->
      <RATE_OF_TURN>Right 5</RATE_OF_TURN> <!-- Right or Left, 0–720
degrees per minute -->
      <SPEED_OVER_GROUND>25.7 Knots</SPEED_OVER_GROUND> <!-- 0 to 102
Knots, 0.1 increments -->
      <POSITION_ACCURACY>3 Meters</POSITION_ACCURACY>
      <LATITUDE>25.095549</LATITUDE>
      <LONGITUDE>-137.625732</LONGITUDE>
      <COURSE_OVER_GROUND>57.6</COURSE_OVER_GROUND> <!-- relative to
true N to 0.1 degree -->
          <TRUE_HEADING>57.8</TRUE_HEADING>    <!-- 0 to 359 degrees
from gyro compass  -->
      <TIME_STAMP>9/28/08 14:17</TIME_STAMP>    <!-- UTC time stamp for
data -->

<!-- AIS broadcasts the following data every 6 minutes
-->
        <IMO_NUMBER>IMO 7654321</IMO_NUMBER>  <!-- # unchanged upon
transfer of registry -->
        <RADIO_CALL_SIGN>CGA2X54</RADIO_CALL_SIGN>  <!-- up to seven
characters -->
        <TYPE_POSITIONING>LORAN-C</TYPE_POSITIONING> <!-- GPS, DGPS,
LORAN-C -->
    <LENGTH>250</LENGTH>
        <WIDTH>32</WIDTH>
    <DRAUGHT>30</DRAUGHT>
    <DESTINATION_PORT>Not Reported</DESTINATION_PORT>  <!-- Max 20
characters -->
    <EST_ARRIVAL>0/0/00 00:00</EST_ARRIVAL>  <!-- UTC / month/date/year
hour:minute -->
    <CLASSIFICATION>Unclassified</CLASSIFICATION>
    <ALERT_PRESENT>No</ALERT_PRESENT>

</VESSEL>  <!--  -->

  <VESSEL>
    <!-- Sample tracking data for vessels
-->
    <NAME_TXT>Name: </NAME_TXT>
    <NAME>Globalstar2</NAME>
    <MMSI_TXT>MMSI: </MMSI_TXT>
    <REGISTRY>China</REGISTRY>
    <MILITARY>No</MILITARY>
    <ORIGINATING_PORT>Shanghai</ORIGINATING_PORT>
    <TYPE>Commercial Vessel</TYPE>
    <SUB_TYPE>Container Ship</SUB_TYPE>
    <DEPARTURE>9/15/08 02:45</DEPARTURE>
    <!-- UTC / month/date/year hour:minute -->

    <!-- AIS sends the following data every 2–10 seconds while underway
       depending on speed, and every 3 minutes while at anchor
-->
    <MMSI>412159197</MMSI>
    <!-- Maritime Mobile Service Identity 412, 413 China-->
    <NAV_STATUS>Under Way Using Engines</NAV_STATUS>
    <!-- At Anchor, Under Way, etc. -->
    <RATE_OF_TURN>0</RATE_OF_TURN>
    <!-- Right or Left, 0–720 degrees per minute -->
    <SPEED_OVER_GROUND>11.2 Knots</SPEED_OVER_GROUND>
    <!-- 0 to 102 Knots, 0.1 increments -->
    <POSITION_ACCURACY>3 Meters</POSITION_ACCURACY>
    <LATITUDE>47.147145</LATITUDE>
    <LONGITUDE>-121.342285</LONGITUDE>
    <COURSE_OVER_GROUND></COURSE_OVER_GROUND>
    <!-- relative to true N to 0.1 degree -->
    <TRUE_HEADING>67</TRUE_HEADING>
    <!-- 0 to 359 degrees from gyro compass  -->
    <TIME_STAMP>9/28/08 09:49</TIME_STAMP>
    <!-- UTC time stamp for data -->

    <!-- AIS broadcasts the following data every 6 minutes
-->
    <IMO_NUMBER>IMO 1234561</IMO_NUMBER>
    <!-- # unchanged upon transfer of registry -->
    <RADIO_CALL_SIGN>G2CS1</RADIO_CALL_SIGN>
    <!-- up to seven characters -->
    <TYPE_POSITIONING>LORAN-C</TYPE_POSITIONING>
    <!-- GPS, DGPS, LORAN-C -->
    <LENGTH>200</LENGTH>
    <WIDTH>25</WIDTH>
    <DRAUGHT>25</DRAUGHT>
    <DESTINATION_PORT>San Diego</DESTINATION_PORT>
    <!-- Max 20 characters -->
    <EST_ARRIVAL>10/2/08 15:00</EST_ARRIVAL>
    <!-- UTC / month/date/year hour:minute -->
    <CLASSIFICATION>Unclassified</CLASSIFICATION>
    <ALERT_PRESENT>No</ALERT_PRESENT>

  </VESSEL>
  <!--  -->
  <VESSEL>
    <!-- Sample tracking data for vessels
-->
    <NAME_TXT>Name: </NAME_TXT>
    <NAME>Globalstar8</NAME>
    <MMSI_TXT>MMSI: </MMSI_TXT>
    <REGISTRY>China</REGISTRY>
    <MILITARY>No</MILITARY>
    <ORIGINATING_PORT>San Diego</ORIGINATING_PORT>
    <TYPE>Commercial Vessel</TYPE>
    <SUB_TYPE>Container Ship</SUB_TYPE>
    <DEPARTURE>9/27/08 21:15</DEPARTURE>
    <!-- UTC / month/date/year hour:minute -->

    <!-- AIS sends the following data every 2–10 seconds while underway
       depending on speed, and every 3 minutes while at anchor
-->
    <MMSI>412159178</MMSI>
    <!-- Maritime Mobile Service Identity 412, 413 China-->
    <NAV_STATUS>Under Way Using Engines</NAV_STATUS>
    <!-- At Anchor, Under Way, etc. -->
    <RATE_OF_TURN>0</RATE_OF_TURN>
    <!-- Right or Left, 0–720 degrees per minute -->
    <SPEED_OVER_GROUND>15.6 Knots</SPEED_OVER_GROUND>
    <!-- 0 to 102 Knots, 0.1 increments -->
    <POSITION_ACCURACY>3 Meters</POSITION_ACCURACY>
    <LATITUDE>37.147145</LATITUDE>
    <LONGITUDE>-18.342285</LONGITUDE>
    <COURSE_OVER_GROUND></COURSE_OVER_GROUND>
    <!-- relative to true N to 0.1 degree -->
    <TRUE_HEADING>98</TRUE_HEADING>
    <!-- 0 to 359 degrees from gyro compass  -->
    <TIME_STAMP>9/28/08 14:45</TIME_STAMP>
    <!-- UTC time stamp for data -->

    <!-- AIS broadcasts the following data every 6 minutes
-->
    <IMO_NUMBER>IMO 1234568</IMO_NUMBER>
    <!-- # unchanged upon transfer of registry -->
    <RADIO_CALL_SIGN>G8CS1</RADIO_CALL_SIGN>
    <!-- up to seven characters -->
    <TYPE_POSITIONING>LORAN-C</TYPE_POSITIONING>
    <!-- GPS, DGPS, LORAN-C -->
    <LENGTH>200</LENGTH>
    <WIDTH>25</WIDTH>
    <DRAUGHT>25</DRAUGHT>
    <DESTINATION_PORT>Oakland</DESTINATION_PORT>
    <!-- Max 20 characters -->
    <EST_ARRIVAL>10/5/08 19:30</EST_ARRIVAL>
    <!-- UTC / month/date/year hour:minute -->
    <CLASSIFICATION>Unclassified</CLASSIFICATION>
    <ALERT_PRESENT>Yes</ALERT_PRESENT>

  </VESSEL>
  <!--  -->

</VESSELS>
```

Listing 1. XML Code for Vessels.xml

This code represents a small-scale sample of the type of data that might be available in a production level Radiant Alloy system that is connected into real-time repositories and reporting systems based on the National Information Exchange Model (NIEM) [9]. By using this sample XML code as a foundation, we can test a live query based request for data and the integration of the ID used in the re-architecting of our system with existing alerts.

1. Alert Messages

System Alert Notifications are added to the replicated domain-level IBs as the notifications are archived and retrieved. Alert Notification messages are based on the standard maritime Advance Notice of Arrival (ANOA) and use the XML-based Maritime Information Exchange Model (MIEM) [5]. The alert includes fields from a standard Vessel Activity Report (VAR) to show details such as identification, kinematics, and port history, as well as the specific instructions pertaining to the alert. In our scenario, the EWO processes an Alert Notification about the Globalstar7, as he or she updates some part of the VAR data for that vessel, thus causing the VAR to “disappear” from the lower level (Unclassified) view in the system.

These Alert messages become a critical piece of the Use-Case scenario that was presented. As vessels are processed in the system as Vessels of Interest (VOI), alerts are generated. These alerts are processed at whichever classification level they originate from (e.g., Top Secret), and the role of the IB will be extended to both distribute and query these alerts as the system is used. Since each alert contains information pertaining to a specific vessel, port, and time group, we can effectively query the XML-based information in the same manner as we would any other data. This allows a seamless integration with our ID and our domain-level IBs to process queries based on Vessel and Alert Message details. Our Use-Case scenario dictates that valid Alert Messages will be generated to notify ports or facilities of pertinent information. These Alerts will include the following:

- To Information (a Port or a Role can be used in this field)
- Date Time Group (DTG) of the Alert generation
- Suspense for action on the Alert

- Special instructions or Comments regarding the message

- Vessel of Interest (VOI) information

As the Electronic Warfare Officer (EWO) performs his duties, he is responsible for the generation of Alert Messages. These messages are intended to facilitate port security and homeland defense, and the EWO has the responsibility of transmitting the Alert Message through the system. Listing 2 shows a sample alert message XML file was generated for the High Level Alert Use-Case scenario.

```
<?xml version=“1.0” encoding=“utf-8” ?>
<?xml-stylesheet type=“text/css” href=“alert.css” ?>
<?xml-stylesheet type=“text/css” href=“vessels.css” ?>


<ALERTS>

<TITLE>ALERT Notification</TITLE>
<MESSAGE>
<![CDATA[
       Alert for a Vessel of Interest (VOI).
]]>
</MESSAGE>

<!-- Alert format for Use Case: -->
<ALERT>
      <ALERT_TO>San Diego</ALERT_TO>
      <ALERT_DTG>9/28/08 05:00</ALERT_DTG>
      <SUSPENSE>10/01/08 23:59</SUSPENSE>
      <COMMENTS>
       <![CDATA[

             VOI -- notify local CG Office @ (619) 867-5309 upon arrival and
inspect cargo with CG Duty Officer

             ]]>
      </COMMENTS>
             <VESSEL>
             <!-- Sample tracking data for vessels
             -->
            <NAME_TXT>Name: </NAME_TXT>
            <NAME>Globalstar7</NAME>
               <MMSI_TXT>MMSI: </MMSI_TXT>
            <REGISTRY>China</REGISTRY>
            <MILITARY>No</MILITARY>
            <ORIGINATING_PORT>Shanghai</ORIGINATING_PORT>
            <TYPE>Commercial Vessel</TYPE>
            <SUB_TYPE>Container Ship</SUB_TYPE>
            <DEPARTURE>8/28/08 08:45</DEPARTURE>  <!-- UTC / month/date/year
hour:minute -->

             <!-- AIS sends the following data every 2–10 seconds while
underway
```

```
                    depending on speed, and every 3 minutes while at anchor
             -->
            <MMSI>412159177</MMSI>  <!-- Maritime Mobile Service Identity 412,
413 China-->
            <NAV_STATUS>Under Way Using Engines</NAV_STATUS> <!-- At Anchor,
Under Way, etc. -->
            <RATE_OF_TURN>0</RATE_OF_TURN> <!-- Right or Left, 0–720 degrees
per minute -->
            <SPEED_OVER_GROUND>10.2 Knots</SPEED_OVER_GROUND> <!-- 0 to 102
Knots, 0.1 increments -->
            <POSITION_ACCURACY>3 Meters</POSITION_ACCURACY>
            <LATITUDE>27.147145</LATITUDE>
            <LONGITUDE>-128.342285</LONGITUDE>
            <COURSE_OVER_GROUND></COURSE_OVER_GROUND> <!-- relative to true N
to 0.1 degree -->
               <TRUE_HEADING>27</TRUE_HEADING>    <!-- 0 to 359 degrees from
gyro compass  -->
            <TIME_STAMP></TIME_STAMP>    <!-- UTC time stamp for data -->

             <!-- AIS broadcasts the following data every 6 minutes
             -->
               <IMO_NUMBER>IMO 1234567</IMO_NUMBER>  <!-- # unchanged upon
transfer of registry -->
               <RADIO_CALL_SIGN>G7CS1</RADIO_CALL_SIGN>  <!-- up to seven
characters -->
               <TYPE_POSITIONING>LORAN-C</TYPE_POSITIONING> <!-- GPS, DGPS,
LORAN-C -->
            <LENGTH>200</LENGTH>
               <WIDTH>25</WIDTH>
            <DRAUGHT>25</DRAUGHT>
            <DESTINATION_PORT>San Diego</DESTINATION_PORT>  <!-- Max 20
characters -->
            <EST_ARRIVAL>10/1/08 13:00</EST_ARRIVAL>  <!-- UTC /
month/date/year hour:minute -->
        <CLASSIFICATION>Top Secret</CLASSIFICATION>
        <ALERT_PRESENT>Yes</ALERT_PRESENT>

    </VESSEL>  <!--   -->
</ALERT>

  <!-- Alert format for Use Case: -->
  <ALERT>
    <ALERT_TO>Oakland</ALERT_TO>
    <ALERT_DTG>10/02/08 05:00</ALERT_DTG>
    <SUSPENSE>10/06/08 23:59</SUSPENSE>
    <COMMENTS>
      <![CDATA[

             VOI -- call Dr. Falken from Protovision at 555–8632, also alert
INS and local CG

             ]]>
    </COMMENTS>
    <VESSEL>
      <!-- Globalstar8
-->
      <NAME_TXT>Name: </NAME_TXT>
      <NAME>Globalstar8</NAME>
      <MMSI_TXT>MMSI: </MMSI_TXT>
      <REGISTRY>China</REGISTRY>
      <MILITARY>No</MILITARY>
      <ORIGINATING_PORT>San Diego</ORIGINATING_PORT>
      <TYPE>Commercial Vessel</TYPE>
      <SUB_TYPE>Container Ship</SUB_TYPE>
      <DEPARTURE>9/27/08 21:15</DEPARTURE>
      <!-- UTC / month/date/year hour:minute -->

      <!-- AIS sends the following data every 2–10 seconds while underway
       depending on speed, and every 3 minutes while at anchor
-->
      <MMSI>412159178</MMSI>
      <!-- Maritime Mobile Service Identity 412, 413 China-->
      <NAV_STATUS>Under Way Using Engines</NAV_STATUS>
      <!-- At Anchor, Under Way, etc. -->
      <RATE_OF_TURN>0</RATE_OF_TURN>
      <!-- Right or Left, 0–720 degrees per minute -->
      <SPEED_OVER_GROUND>15.6 Knots</SPEED_OVER_GROUND>
      <!-- 0 to 102 Knots, 0.1 increments -->
      <POSITION_ACCURACY>3 Meters</POSITION_ACCURACY>
      <LATITUDE>37.147145</LATITUDE>
      <LONGITUDE>-18.342285</LONGITUDE>
      <COURSE_OVER_GROUND></COURSE_OVER_GROUND>
      <!-- relative to true N to 0.1 degree -->
      <TRUE_HEADING>98</TRUE_HEADING>
      <!-- 0 to 359 degrees from gyro compass  -->
      <TIME_STAMP>9/28/08 14:45</TIME_STAMP>
      <!-- UTC time stamp for data -->

      <!-- AIS broadcasts the following data every 6 minutes
-->
      <IMO_NUMBER>IMO 1234568</IMO_NUMBER>
      <!-- # unchanged upon transfer of registry -->
      <RADIO_CALL_SIGN>G8CS1</RADIO_CALL_SIGN>
      <!-- up to seven characters -->
      <TYPE_POSITIONING>LORAN-C</TYPE_POSITIONING>
      <!-- GPS, DGPS, LORAN-C -->
      <LENGTH>200</LENGTH>
      <WIDTH>25</WIDTH>
      <DRAUGHT>25</DRAUGHT>
      <DESTINATION_PORT>Oakland</DESTINATION_PORT>
      <!-- Max 20 characters -->
      <EST_ARRIVAL>10/5/08 19:30</EST_ARRIVAL>
      <!-- UTC / month/date/year hour:minute -->
      <CLASSIFICATION>Unclassified</CLASSIFICATION>
      <ALERT_PRESENT>Yes</ALERT_PRESENT>

    </VESSEL>
    <!--  -->
  </ALERT>
</ALERTS>
```

Listing 2. XML Code for Alerts.xml

Despite having only two alerts in the sample data store, we can still show the ability to parse through the data for alerts that pertain to specific locations and provide a basis for the testing of our query engine and the resultant ID rules that are created to specify our security policy. In our High Level Alert Use-Case scenario, we find that the

EWO is at a higher classification level (i.e., Top Secret), than the Harbormaster to whom the Alert Message is intended. Since the EWO is operating in a different domain than the Harbormaster, when he touches information in the system, by marking the Globalstar7 as a VOI, he effectively prevents all lower classification users from seeing this track (due to the tranquilty property of BLP). This is where the unintended creation of information leakage arises when the Harbormaster becomes alerted to the presence of a ship in the system that he cannot access information pertaining to from his resultant queries.

B. QUERIES

Several methods exist for generating queries and controlling access to data stores based on XML. These include XQuery and XPath, developed by the W3C [67]. XMLQuery, however, does not allow the modification of data and subsequent saving of the modified information back into the data store and is a deprecated version of XPath. User queries are a standard and integral service of the IB and are generated using Xpath. Each request for data (i.e., query) processed by the IB is first checked for permissions by the Policy Decision Point (PDP) service, then passed via a service call to the ID service associated with that classification level. The IB then begins execution of the query itself but must wait for a positive or negative response from the ID service before returning results to the user. Valid queries can be performed on any field available from an Alert Message or within a VAR. A sample Xpath query for the San Diego Harbormaster requesting a Destination Port query is shown in Listing 3.

```
/VESSELS/VESSEL/DESTINATION_PORT[text()=‘San Diego’]
```

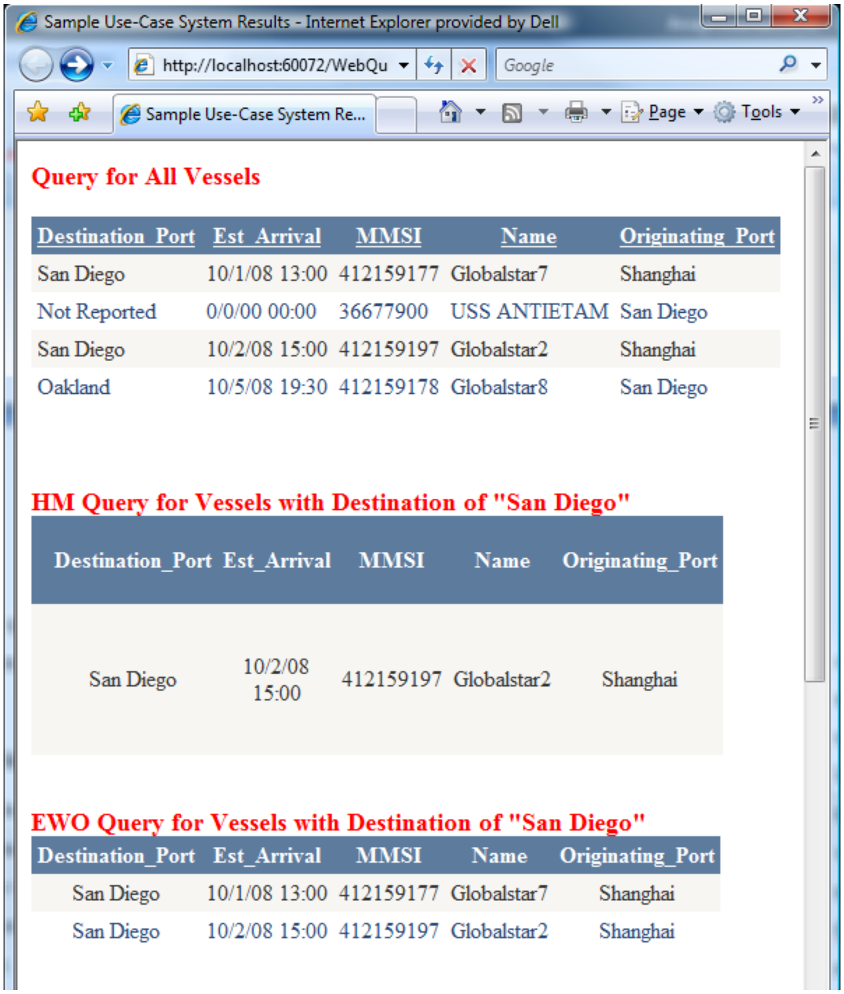
Listing 3. Xpath Query for San Diego Harbormaster using Destination Port

Integral to this query are the attributes for the user (role, location, security level, and time stamp) that are also passed with the query for use by both the IB and ID services. These attached attributes are critical to the IB’s response since the XPath query that is used for the EWO (Listing 4) is identical to the query from the Harbormaster.

```
/VESSELS/VESSEL/DESTINATION_PORT[text()=‘San Diego’]
```

Listing 4. Xpath Query for EWO using Destination Port of San Diego

The differences in the query return data are from the IB’s use of those additional attributes attached to the initial query. Figure 22 shows the XML data present for all vessels in the system and then the expected results from San Diego destination port queries for both the Harbormaster and the EWO.


Sample Use-Case System Results - Internet Explorer provided by Dell
http://localhost:60072/WebQu
Google
Sample Use-Case System Re...
Page
Tools

Query for All Vessels

Destination_Port	Est_Arrival	MMSI	Name	Originating_Port
San Diego	10/1/08 13:00	412159177	Globalstar7	Shanghai
Not Reported	0/0/00 00:00	36677900	USS ANTIETAM	San Diego
San Diego	10/2/08 15:00	412159197	Globalstar2	Shanghai
Oakland	10/5/08 19:30	412159178	Globalstar8	San Diego

HM Query for Vessels with Destination of "San Diego"

Destination_Port	Est_Arrival	MMSI	Name	Originating_Port
San Diego	10/2/08 15:00	412159197	Globalstar2	Shanghai

EWO Query for Vessels with Destination of "San Diego"

Destination_Port	Est_Arrival	MMSI	Name	Originating_Port
San Diego	10/1/08 13:00	412159177	Globalstar7	Shanghai
San Diego	10/2/08 15:00	412159197	Globalstar2	Shanghai

Figure 22. Expected XPath Query Results by Role

In our High Level Alert Use-Case scenario, valid queries can be performed on any field available from an Alert Message or within Vessel details. The data set that the system returns must be restricted based upon Role permission for each of our actors in the scenario. In this case, the query and resultant data must include the Harbormaster’s port (either Origination or Destination) for details to be returned by the IB and ID. This will prevent a Harbormaster from just querying the system to get information about vessels from the system that do not directly pertain to his role (at his location) and his direct performance of duty. But, he should be entitled to know information about ships arriving or departing from his location, since this is a key responsibility of his job. Additionally, the Harbormaster should be able to query the system for the presence of Alert Messages that pertain to his role. These aspects are areas that must be enforced using the IB, but can only be done as a result of our re-architecting through the addition of an ID and its rule processing.

This result is reflective of a standard query to the system and can be extended as necessary to support other scenarios. The XPath query can be used on both standard vessel data that is present in the system, as well as for Alert Messages that are present. Listing 5 depicts a sample Xpath query for Alert Messaging for the port of San Diego.

```
 /ALERTS/ALERT/DESTINATION_PORT[text()=‘San Diego’]
```

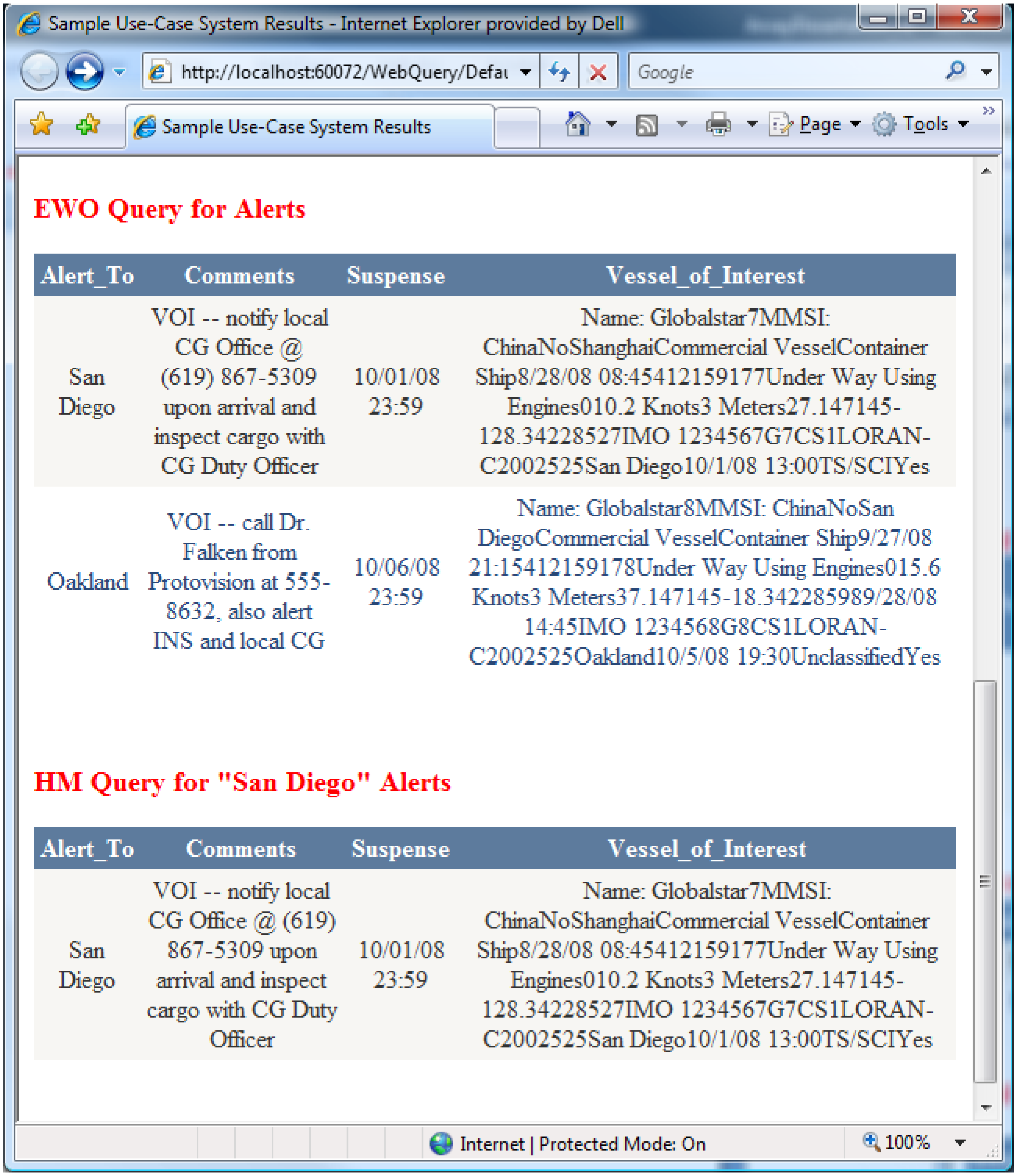
Listing 5. Xpath Alert Messaging Query for San Diego Harbormaster

While Listing 6 depicts the query that would be executed by the EWO for Alerts present at the TS level. This query would return all alerts present at his security level, since we are not restricting the EWO’s access because of his role.

```
/ALERTS/ALERT
```

Listing 6. Xpath Alert Messaging Query for EWO

The expected results of both of these Alert Messaging queries are shown in Figure 23.


Sample Use-Case System Results - Internet Explorer provided by Dell
http://localhost:60072/WebQuery/Defau
Google
Sample Use-Case System Results
Page
Tools
Internet | Protected Mode: On
100%

EWO Query for Alerts

Alert_To	Comments	Suspense	Vessel_of_Interest
San Diego	VOI -- notify local CG Office @ (619) 867-5309 upon arrival and inspect cargo with CG Duty Officer	10/01/08 23:59	Name: Globalstar7MMSI: ChinaNoShanghaiCommercial VesselContainer Ship8/28/08 08:45412159177Under Way Using Engines010.2 Knots3 Meters27.147145-128.34228527IMO 1234567G7CS1LORAN-C2002525San Diego10/1/08 13:00TS/SCIYes
Oakland	VOI -- call Dr. Falken from Protovision at 555-8632, also alert INS and local CG	10/06/08 23:59	Name: Globalstar8MMSI: ChinaNoSan DiegoCommercial VesselContainer Ship9/27/08 21:15412159178Under Way Using Engines015.6 Knots3 Meters37.147145-18.342285989/28/08 14:45IMO 1234568G8CS1LORAN-C2002525Oakland10/5/08 19:30UnclassifiedYes

HM Query for "San Diego" Alerts

Alert_To	Comments	Suspense	Vessel_of_Interest
San Diego	VOI -- notify local CG Office @ (619) 867-5309 upon arrival and inspect cargo with CG Duty Officer	10/01/08 23:59	Name: Globalstar7MMSI: ChinaNoShanghaiCommercial VesselContainer Ship8/28/08 08:45412159177Under Way Using Engines010.2 Knots3 Meters27.147145-128.34228527IMO 1234567G7CS1LORAN-C2002525San Diego10/1/08 13:00TS/SCIYes

Figure 23. Expected XPath Query Alert Messaging Results by Role

As an additional query method and to better support a web-enable test framework for query integration, we chose to utilize Language Integrated Query (LINQ) and integrate the queries using Visual Basic. It includes query, set, and transform operations via integrated library functions for data. LINQ to XML takes advantage of standard query operators and adds query extensions specific to XML, as well as offering integration with RDF format data. Either of these query methods (XPath or LINQ) can exist in Radiant Alloy. The web page version could easily be modified to allow for easier manipulation of queries and access to the data, but in this example we relied on hard-coding of queries and singular executions to show that the query system is operational. A sample of the query system using LINQ is shown in Listing 7.

```
Imports System.Xml.Linq

Partial Class _Default
    Inherits System.Web.UI.Page
    Private xmlDocument As XDocument
    Protected Overrides Sub OnLoad(ByVal e As System.EventArgs)
        MyBase.OnLoad(e)
        LoadXMLDocument()
        GridView1.DataSource = DescendantsQuery1()
        GridView1.DataBind()
        GridView2.DataSource = DescendantsQuery2()
        GridView2.DataBind()
        Dim vesselList1 As XDocument =
XDocument.Load(MapPath(“vessels.xml”))
        Dim alertList1 As XDocument =
XDocument.Load(MapPath(“alerts.xml”))

        GridView3.DataSource = From VESSEL In vesselList1...<VESSEL>
Where VESSEL.<ORIGINATING_PORT>.Value = “San Diego” And
VESSEL.<CLASSIFICATION>.Value = “Unclassified” Select Name =
VESSEL.<NAME>.Value, MMSI = VESSEL.<MMSI>.Value, Est_Arrival =
VESSEL.<EST_ARRIVAL>.Value, Originating_Port =
VESSEL.<ORIGINATING_PORT>.Value, Destination_Port =
VESSEL.<DESTINATION_PORT>.Value
        GridView3.DataBind()
        GridView5.DataSource = From VESSEL In vesselList1...<VESSEL>
Where VESSEL.<DESTINATION_PORT>.Value = “San Diego” And
(VESSEL.<CLASSIFICATION>.Value = “Top Secret” Or
VESSEL.<CLASSIFICATION>.Value = “Unclassified”) Select Name =
VESSEL.<NAME>.Value, MMSI = VESSEL.<MMSI>.Value, Est_Arrival =
VESSEL.<EST_ARRIVAL>.Value, Originating_Port =
VESSEL.<ORIGINATING_PORT>.Value, Destination_Port =
VESSEL.<DESTINATION_PORT>.Value
        GridView5.DataBind()

        GridView4.DataSource = From ALERT In alertList1...<ALERT>
Select Alert_To = ALERT.<ALERT_TO>.Value, Suspense =
ALERT.<SUSPENSE>.Value, Comments = ALERT.<COMMENTS>.Value,
Vessel_of_Interest = ALERT.<VESSEL>.Value
        GridView4.DataBind()

        GridView6.DataSource = From ALERT In alertList1...<ALERT> Where
ALERT.<ALERT_TO>.Value = “San Diego” And
```

```
(ALERT.<VESSEL>.<ORIGINATING_PORT>.Value = “San Diego” Or
ALERT.<VESSEL>.<DESTINATION_PORT>.Value = “San Diego”) Select Alert_To
= ALERT.<ALERT_TO>.Value, Suspense = ALERT.<SUSPENSE>.Value, Comments =
ALERT.<COMMENTS>.Value, Vessel_of_Interest = (ALERT.<VESSEL>.Value)
        GridView6.DataBind()
        ‘‘Demonstrates traversing XML in a LINQ query.
        ‘For Each x In xmlDocument...<VESSELS>.<NAME>
    End Sub
    Public Sub LoadXMLDocument()
        xmlDocument = XDocument.Load(MapPath(“vessels.xml”))
    End Sub

    ‘‘‘ <summary>
    ‘‘‘ Query the vessels descendants for details using Element.
    ‘‘‘ </summary>
    ‘‘‘ <returns>Object</returns>
    Public Function DescendantsQuery1() As Object
        Return From VESSEL In xmlDocument...<VESSEL> Select Name =
VESSEL.<NAME>.Value, MMSI = VESSEL.<MMSI>.Value, Est_Arrival =
VESSEL.<EST_ARRIVAL>.Value, Originating_Port =
VESSEL.<ORIGINATING_PORT>.Value, Destination_Port =
VESSEL.<DESTINATION_PORT>.Value

    End Function

    ‘‘‘ <summary>
    ‘‘‘ Query the vessels descendants for details using Element.
    ‘‘‘ </summary>
    ‘‘‘ <returns>Object</returns>
    Public Function DescendantsQuery2() As Object
        Return From VESSEL In xmlDocument...<VESSEL> Where
VESSEL.<DESTINATION_PORT>.Value = “San Diego” And
VESSEL.<CLASSIFICATION>.Value = “Unclassified” Select Name =
VESSEL.<NAME>.Value, MMSI = VESSEL.<MMSI>.Value, Est_Arrival =
VESSEL.<EST_ARRIVAL>.Value, Originating_Port =
VESSEL.<ORIGINATING_PORT>.Value, Destination_Port =
VESSEL.<DESTINATION_PORT>.Value

    End Function
End Class
```

Listing 7. Language Integrated Query Sample Code

The results produced from this framework are identical to those produced using XPath as the query language and were already shown (Figure 22 and Figure 23). Based on the integration of the query system with the existing vessel data sets, we can begin to develop rules for our ID to support the re-architecting of the system and the specification of our security policy using RuleML.

VI. THE INFORMATION DECLASSIFIER

In support of the re-architecting effort based on the UML Use-Case analysis, we have added the concept of an Information Declassifier. The original process flow of Radiant Alloy relied on queries to the IB for direct results from associated repositories as shown in Figure 24.

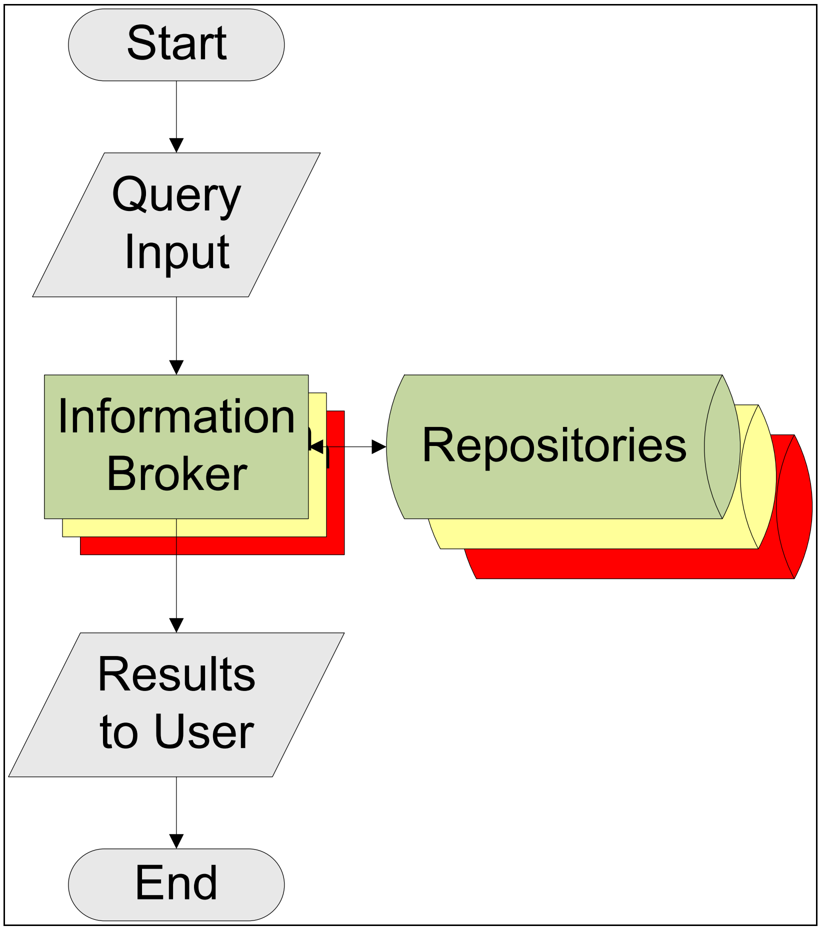


Figure 24. General Process Flow for Radiant Alloy

The revised process flow resulting from the rearchitecting process now includes the ID as an element of the architecture is shown in Figure 25. The ID is specified using RuleML and was required to satisfy the security use-case that was depicted in Figure 18 and Figure 20.

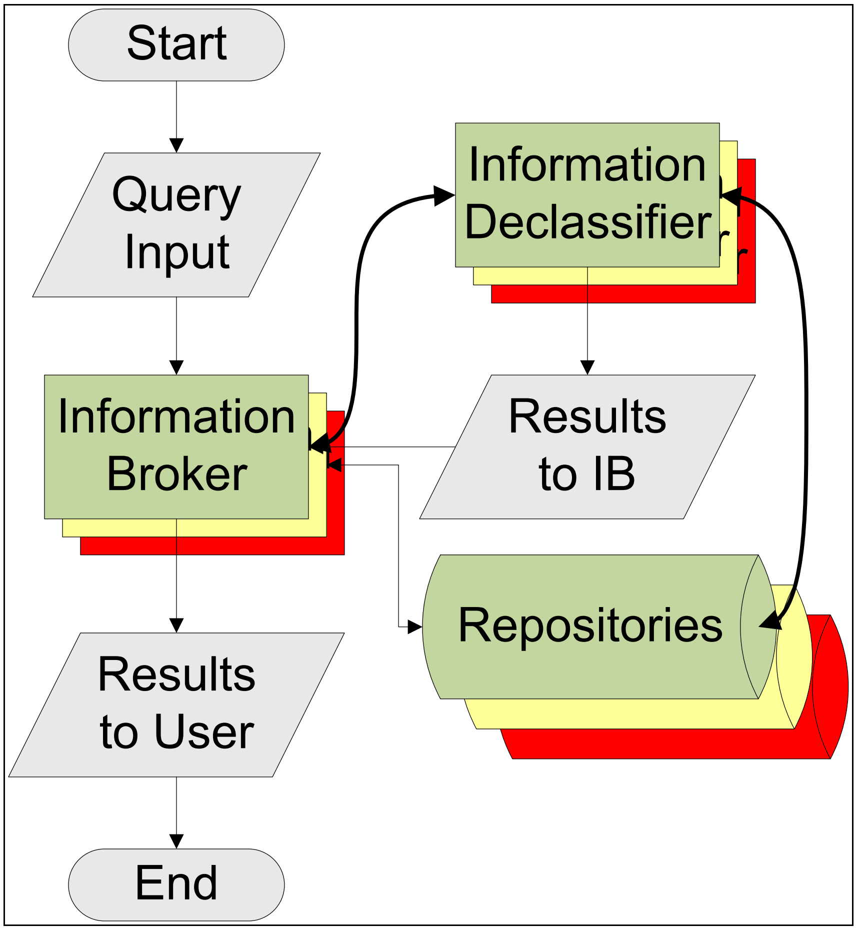


Figure 25. Revised Process Flow for Modified Radiant Alloy

The high-level operational vision for the ID and IB interaction can be explained as follows. The IB is the mechanism for user entry to the system and executing queries. The IB is also responsible for initiating the ID service and passing both user attributes and query data to the ID. The ID, which is replicated at all security domains in the MLS system, then begins its rule processing. The logical flow of IB and ID interaction is shown in Figure 26.

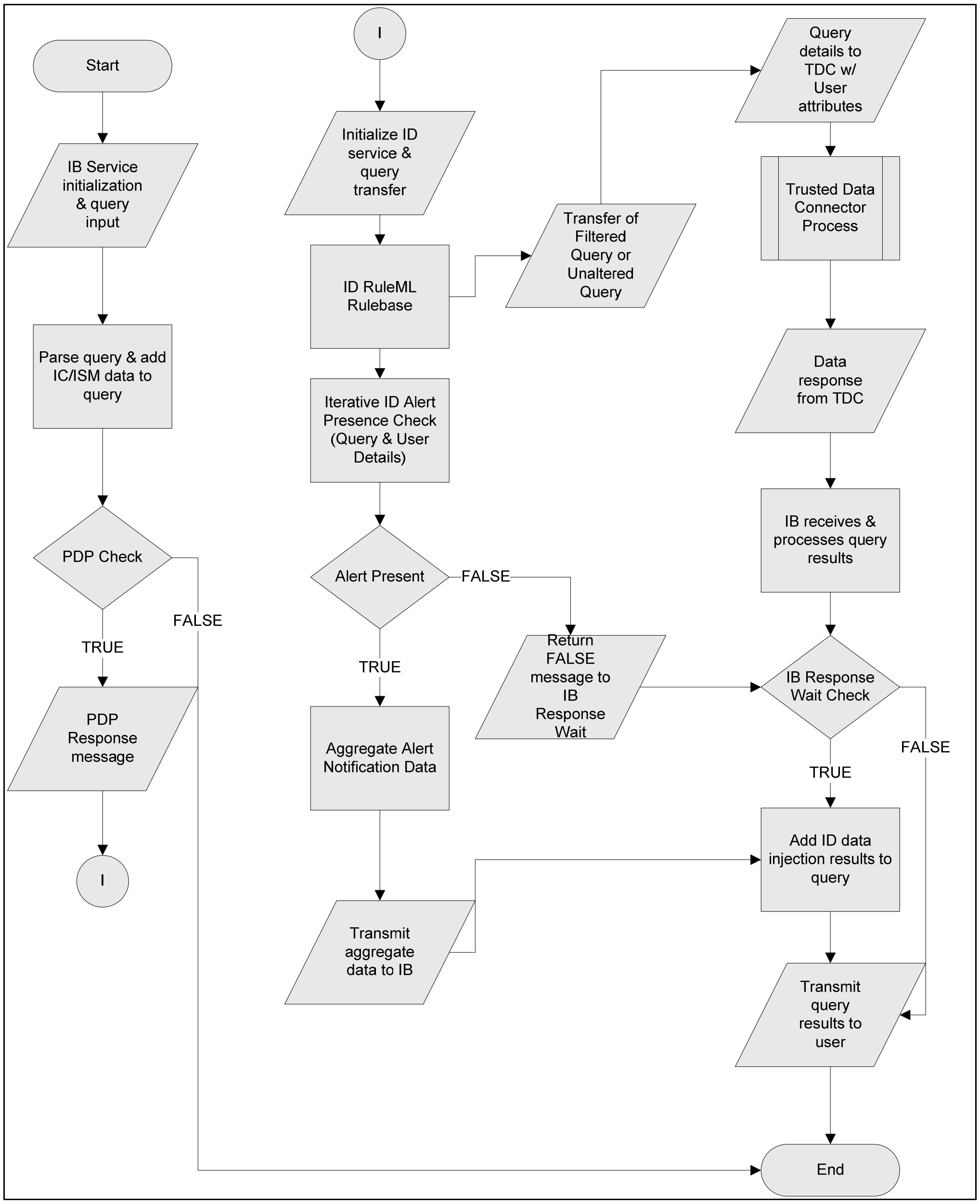

Start
IB Service initialization & query input
Parse query & add IC/ISM data to query
PDP Check
TRUE
FALSE
PDP Response message
I
I
Initialize ID service & query transfer
ID RuleML Rulebase
Iterative ID Alert Presence Check (Query & User Details)
Alert Present
FALSE
TRUE
Aggregate Alert Notification Data
Transmit aggregate data to IB
Transfer of Filtered Query or Unaltered Query
Return FALSE message to IB Response Wait
Query details to TDC w/ User attributes
Trusted Data Connector Process
Data response from TDC
IB receives & processes query results
IB Response Wait Check
TRUE
FALSE
Add ID data injection results to query
Transmit query results to user
End

Figure 26. Flow Chart of the Operational Vision

The parallelograms represent input or messaging, the rectangles represent processing, the diamonds represent decisions, and the circles are on-page continuations of flow. This flow chart does not show the iterative ID process that must occur for every IB query. The ID is also responsible for initiating the higher level ID services and passing the same parameters to the higher level ID (until the top domain is reached). Beyond this operational vision we must rely on the individual rule set to determine our policy and implement its functionality. The entire rule set to support the Use Case of our analysis comprises approximately forty separate rules that use forward chaining, thereby allowing higher level rules to link to subordinate rules or variables to link fields in authorized queries to the system, as well as data classification and user security levels to form the basis for our evaluative process.

Additionally, we have re-routed queries from the existing architecture to account for the presence of the ID and its role in the system. The Information Declassifier must act as an enforcer for information flows that may produce unwanted leakage. This leakage could be from disparate domains, or even within a domain but between Roles and to which a particular subject is not authorized the information that they obtain. These rules that we will describe and enact will be generated in RuleML. They can be executed on any XML specified data or code and are based on our High-Level Alert Use-Case scenario. Three cases must be addressed by the RuleML and Information Declassifier for the re-architecting:

1. Reroute all User-IB queries and Inter-IB queries–queries must be routed to the ID and passed to the highest level ID/IB to allow for the inclusion of data from alert messages that are no longer visible at the requestor's classification level. This includes both intra-domain and inter-domain interaction.

2. Reroute all Alert Notification Messaging–Alerts must be routed through the system (we cannot control out-of-band), but we must provide for a distribution method within the system through the ID.

3. Reroute all Repository Responses–Data responses must be routed through the IB / ID pair, to verify appropriately for action against the rule engine. This will help in efforts of need to share and need not to share while providing support to allow for information filtering.

The Information Broker implementation must be extended to contain an orchestration mechanism to invoke the ID. Similar to the rules for a security kernel, we must ensure that the orchestration mechanism in the IB service is always invoked and tamperproof. Whenever a request or query reaches the IB service, the orchestration will invoke the ID and pass the same informational query, along with the user credentials and role details in the messaging. The IB will continue to execute its primary objective in gathering data from the associated data repositories, but it will also wait for a response from the ID service that was invoked. There will be two cases of response from the ID. A negative response from the ID will result in the IB returning whatever data it found initially, with respect to the role-mapped permissions for the user. A positive response from the ID will cause the IB to augment its results with the additional information that the ID returned.

A means to mitigate risk is to create and derive the rule base that will be used by the ID. Depending upon the most critical information sharing concerns and the need to prevent specified misuse, we will create a rule base that will isolate and address those areas. This will never be all inclusive, nor will each of the cases be mutually exclusive. Certain information sharing rules will overtly conflict with others and will have to be reconciled to achieve an acceptable result for information flow. This ambiguity must be reconciled in the development of the overall security policy. This support of the re-architecting effort based on the UML Use-Case will help to ensure that we maintain our security property for the emergent behaviors that are most critical within our information system, based on the expressed operational context used in this research.

A. RULES FOR THE INFORMATION DECLASSIFIER

RuleML's expressive power offers the potential to implement a rule engine for an ID that is robust enough to function in an MLS environment, yet still flexible enough to allow for intricate rule usage. The rules generated for a RuleML-based ID must be reactionary; much like the rules for an Access Control List (ACL) on a router. The base set of rules is reactionary and, once the requisite functionality is enabled, can be augmented with derivation and transformational rules. These rules can be replicated between domains and changed readily, yet they still have sufficient expressive power to specify the AIS security policy.

Precise translation from an English language security policy to a specification for that same policy expressed with RuleML is difficult. In support of the re-architecting, and in order to ensure the soundness and completeness of our ID implementation using RuleML, two main categories were addressed to meet our security use case. Intra-domain rules were necessary to provide for filtering of information within a domain, while inter-domain queries allow for the injection of data necessary when Alert Messages are present in the system and data must be added to avoid the loss of safety. RuleML allows for predicates and rules to be sourced either internally (within a service) or externally (a call to a separate service). In the case of our ID, we highlighted the Reaction RuleML side to produce a positive response from our ruleset. The entire developed ruleset consists of forty-one rules which will be covered in detail. Some of these rules were created to replicate other services and responses within our envisioned system that were both beyond the scope and could not easily be replicated for this research. The RuleML set can be expressed in Backus-Naur form (BNF) and the trace through the ruleset can be completed as though the entire ruleset were a context-free grammar (CFG).

The predicates used for the creation of the ruleset are shown in Listing 8. These mappings indicate possible values utilized for the MDA scenario and the header code is included in Listing 9.

Alert_Present_Response → “True” | “False” (Currently User defined – would be IB service response)

Classification_Level → “Unclassified” | “Secret” | “Top Secret”

Data_Classification_Level → “Unclassified” | “Secret” | “Top Secret”

EWO_Valid_Query → User_Role && User_Security_level && PDP_Decision

HM_Queries → “Originating Port” | “Destination Port”

HM_Valid_Query → Oakland Harbormaster query is valid destination | Oakland Harbormaster query is valid origination | SD Harbormaster query is valid destination | SD Harbormaster query is valid origination

IB_Classification_Level → “Unclassified” | “Secret” | “Top Secret”

IB_Response_Wait → Valid_Query && Query_Requires_Data_Insertion

PDP_Decision → User_Role_Permissions

Query_Requires_Data_Insertion → Alert_Present_Response

Return_IB_Results → “Vessel Results from Unclassified IB” | “Vessel Results from Unclassified IB with ID Injection” | “Vessel Results from Secret IB and Unclassified IB” | “Vessel Results from Secret IB and Unclassified IB with ID Injection” | “Vessel Results from Top Secret IB and Secret IB and Unclassified IB”

Security_Level → “Unclassified” | “Secret” | “Top Secret”

User_Role → “EWO” | “Electronic Warfare Officer” | “Harbormaster” | “Harbormaster San Diego” | “Harbormaster Oakland” | “Harbormaster Shanghai”

User_Role_Location → “San Diego” | “Oakland” | “Shanghai” | “Not Reported”

User_Role_Permissions → “True” | “False” (Currently User defined – would be PDP service response)

User_Security_Level → “Unclassified” | “Secret” | “Top Secret”

Valid_Query → HM_Valid_Query | EWO_Valid_Query

Vessel_Classification_Level → User_Role && User_Security_level && PDP_Decision

Vessel_Destination_Port → User_Role_Location

Vessel_Originating_Port → User_Role_Location

Listing 8. RuleML Predicate Listing

The rules created for this scenario were numbered using RuleIDs from zero to forty and encoded after creating a natural language description for the rule. Header data is established to begin the RuleML coding. This includes creating the rule base and establishing the location and naming for rules with the uniform resource identifier (URI).

```
<?xml version=“1.0” encoding=“utf-8”?>
<RuleML xmlns=“http://www.ruleml.org/0.91/xsd”>
  <Rulebase>
    <label>
      <Plex>
        <Expr>
          <Fun
            uri=“dc:title”>
            <Ind>MDA_Scenario_Arvay_current</Ind>
          </Fun>
        </Expr>
        <Expr>
          <Fun
            uri=“dc:author”>
            <Ind>Randy Arvay</Ind>
          </Fun>
        </Expr>
        <Expr>
          <Fun
            uri=“dc:date”>
            <Ind>12/4/2008</Ind>
          </Fun>
        </Expr>
      </Plex>
    </label>
    <!--Rule Policy Information_Declassifier-->
    <!--oid of the rule base / module -->
    <Ind>Information_Declassifier</Ind>
```

Listing 9. RuleML Header Code for Information Declassifier

The rules are constructed to produce a confirmed result, positive or negative, for every query. No implicit deny rule was established for this proof-of-concept ruleset. Each of the rules is detailed in Appendix A and contains summary information for the intent of the rule as well as the actual RuleML code that enacts the rule within the system.

The intended functionality of the information declassifier consists of several parts and rules were written to fulfill those component functions as shown in Figure 27 and the summary of the rule naming is shown in Listing 10.

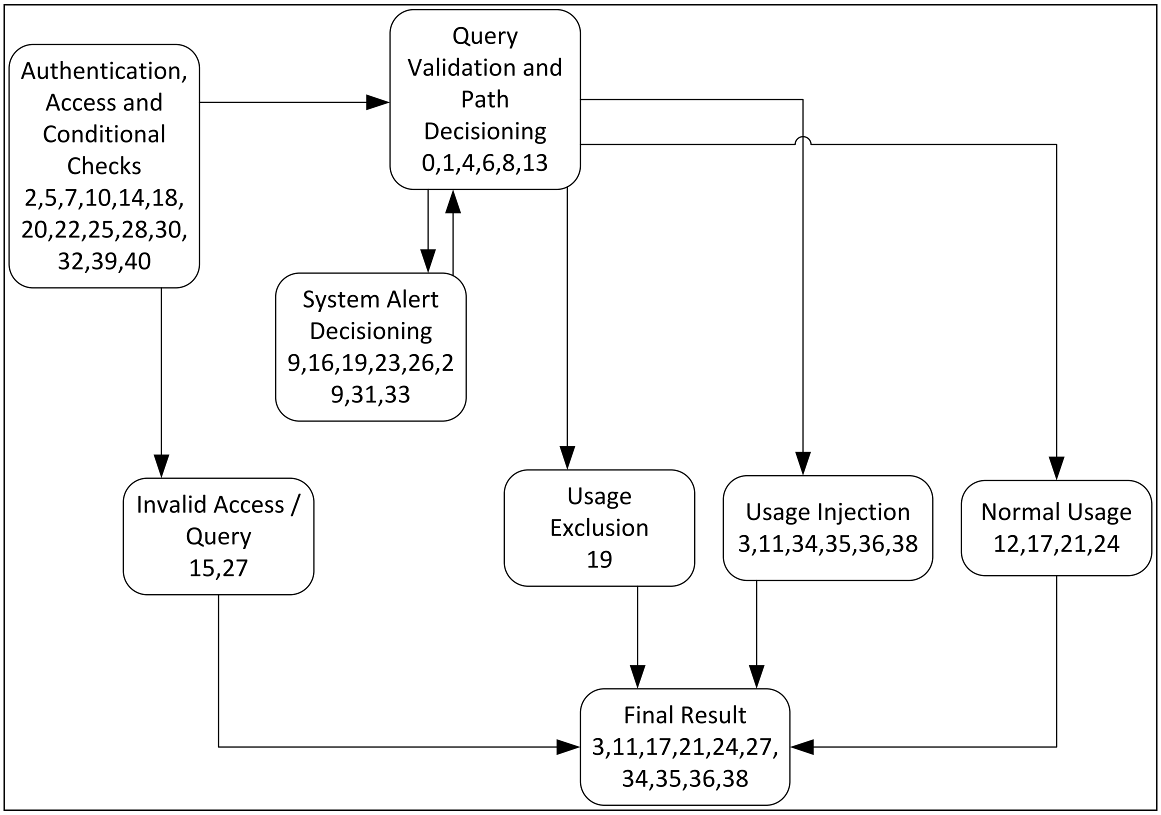


Figure 27. Functional Support for Ruleset

RuleID 0: *Alert Not Valid for Role Location*.

RuleID 1: *Alert Notification is Present*.

RuleID 2: *Est Location for Role*.

RuleID 3: *IB Results 0*.

RuleID 4: *Oakland Harbormaster query is valid destination*.

RuleID 5: *PDP Decision*.

RuleID 6: *Positive ID Response to IB for Alert*.

RuleID 7: *Query is Valid.*

RuleID 8: *Alert Notification is Absent.*

RuleID 9: *Alert SD Secret IB.*

RuleID 10: *EWO Query is Valid*.

RuleID 11: *IB Results 2*.

RuleID 12: *Negative ID Response to IB for Alert*.

RuleID 13: *Oakland Harbormaster query is valid destination 2*.

RuleID 14: *PDP Decision 2*.

RuleID 15: *Query is Invalid.*

RuleID 16: *Alert SD Secret IB 2*.

RuleID 17: *IB Results 3*.

RuleID 18: *Oakland Harbormaster query is valid origination*.

RuleID 19: *Alert SD TS IB.*

RuleID 20: *EWO Query is Valid 2*.

RuleID 21: *IB Results 4*.

RuleID 22: *Oakland Harbormaster query is valid origination 2*.

RuleID 23: *Alert SD TS IB.*

RuleID 24: *IB Results 5*.

RuleID 25: *SD Harbormaster query is valid destination*.

RuleID 26: *Alert Oak Secret IB.*

RuleID 27: *IB Results 6*.

RuleID 28: *SD Harbormaster query is valid destination 2*.

RuleID 29: *Alert Oak Secret IB 2.*

RuleID 30: *SD Harbormaster query is valid origination*.

RuleID 31: *Alert Oak TS IB.*

RuleID 32: *SD Harbormaster query is valid origination 2*.

RuleID 33: *Alert Oak TS IB 2.*

RuleID 34: *IB Results 0.1*.

RuleID 35: *IB Results 0.2*.

RuleID 36: *IB Results 0.3*.

RuleID 37: *No Alert IB.*

RuleID 38: *IB Results 2.1*.

RuleID 39: *EWO Query is Invalid.*

RuleID 40: *HM Query is Invalid.*

Listing 10. RuleML Rule Names

As this proof of concept was not applied to a production system, many conditional rules were required to establish the user credentials and origination data of system queries. These are support rules to allow the validation checks and path determination for the Information Declassifier to operate. With a live production system the use of a Global Content Delivery System (GCDS), or even Microsoft Active Directory in a deprecated case would suffice to provide this level of background and statistical data. Additionally, the checks on alert decisioning were not based on a live repository, so the rules 9, 16, 19, 23, 26, 29, 31, and 33 were all created to replicate this functionality depending on the variable set established in the originating query and sample case. With a live repository and a web application tied into an Oracle (or other) database, these rules would be replaced with a real-time response from that instance. Each of these rules combine to form the basis for expressing our information broker's interaction with a declassifier using RuleML, and demonstrate the interaction required with using a rule engine to exercise these rules.

1. Intra-Domain (Filtering)

By establishing rules that can be enforced within a single information domain, we can effectively limit the ability of users to obtain information to which they are not authorized. One aspect of this intra-domain restriction can be represented by our desire to restrict a Harbormaster's queries to those ports which directly apply to his Role location. Accordingly, if the San Diego Harbormaster wants to query the system for data, then we should only give him the data necessary for the execution of his role. In practice, we want the ID to have a rule that represents the idea that "the Harbormaster's query must only obtain data which includes his Role's port (i.e., San Diego) as either the Origination_Port or the Destination_Port."

2. Inter-Domain (Injection)

The ID must act as an intermediary between varying security domains, represented by an IB at each domain level. This is necessary to process queries from lower classification levels and to pull data pertaining to Alert Message Notifications from higher classification level data stores. When a higher classification or different domain level user touches the data pertaining to a VOI, the lower level user can no longer view that data. This may be counter to their role and responsibilities where they are, in fact, entitled and required to access this information. This information could also be critical to their performance of duties while fulfilling an established Role in the system. The ID would first receive a query message from a domain level IB. This IB is attempting to provide a user with requested data, but in order to prevent an information flow problem as described in our Use-Case scenario, must ask the higher level IB for Alert Message information. This request to a different domain IB, must be processed through a trusted entity, in this case the Information Declassifier that resides between the respective IB domains. As an IB receives a lower level IB query, it must first send a similar request to its higher level (thru the ID between its domains), then it must query its own Alert Message Notification stores for data. Pending the results of the higher level answer and its own Alert Message result, it will generate a Negative Response for No Data Found, or a Positive Response with the associated dataset. The Positive Response will be aggregated to the lower level's own query results, while the Negative Response will result in the IB (at whatever level) processing the query and returning any result it finds (based on user permissions for the Role).

3. Individual Ruleset Descriptions and Purpose

RuleID 0 is titled *Alert Not Valid for Role Location*. This rule is intended to eliminate possibilities for Alert Messaging responses that are not valid for our scenario. If an Alert exists for Los Angeles in the system (at a higher level IB), then this rule toggless that notification to *FALSE*, since it does not apply to the port of San Diego.

RuleID 1 is titled *Alert Notification is Present*. This rule is intended to set the variable *Query_Requires_Data_Insertion* to *TRUE*, if the higher level IB possesses an Alert Message. In this ruleset, the external reference is not made, but invoked through the use of an internally generated variable for the sourcing.

RuleID 2 is titled *Est Location for Role*. This rule is intended to fix the assignment of the user's role location to the results being returned from the IB. This will allow for additional filtering to match the user's location with vessel results from the data repositories that match with origination or destination ports.

RuleID 3 is titled *IB Results 0*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of "Vessel Results from Unclassified IB with ID Injection for Oakland Alert." This is done after checking the user's security level and for the presence of an alert for this location. This rule was created to allow for Harbormaster queries from the port of Oakland as an extension to the San Diego port accounted for in our scenario.

RuleID 4 is titled *Oakland Harbormaster query is valid destination*. This is one of the rules used as an extension to our primary scenario, which would allow for the port of Oakland queries to also work in the system. It is used to verify the user role, the role location, the PDP decision, and the type of query before deciding that it is a valid query and should be processed by the IB.

RuleID 5 is titled *PDP Decision*. This is an externally sourced rule that would be completed by the Policy Decision Point service in our SOA system. It is used, based on the variables it is sourced with, to provide a decision for that user's access to resources in the system. In this implementation, it is not referenced from an actual external PDP service.

RuleID 6 is titled *Positive ID Response to IB for Alert*. This rule is used to indicate a positive response to the presence of an Alert Message at a higher level IB and verifies that the query to the IB is valid based on the roles active in our system. This rule ensures that the IB response that was previously returned directly to the user is now processed through the ID and missing data is injected to the query if necessary, to eliminate the Misuse Case and as a <Prevent> feature of our Security Use Case. This rule is designed to verify other rules that determine if the query to the system is valid, and if the response from the higher level IB requires the system to inject data before processing the query response. It also checks if the query requires data insertion, which is sourced from the higher level IBs. If both of these conditions are *TRUE*, then the IB that received the query is informed that its response to the user must wait for the injection messaging from the ID service prior to returning results to the user.

RuleID 7 is titled *Query is Valid.* This is a filtering rule and is used to verify that the user (EWO or Harbormaster) is requesting an allowable query. It is sourced from two other rules, which check the validity of the query against each actor's allowable query set.

RuleID 8 is titled *Alert Notification is Absent.* This rule is intended to set the variable *Query_Requires_Data_Insertion* to *FALSE*, if the higher level IB does not possess an Alert Message. In this ruleset, the external reference is not made, but invoked through the use of an internally generated variable for the sourcing.

RuleID 9 is titled *Alert SD Secret IB.* This rule is used to verify that when an alert is present at the Secret level IB, intended for San Diego, and the requestor is at the Unclassified security level, that the *Alert_Present_Response* is set to *TRUE.* This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert's classification level.

RuleID 10 is titled *EWO Query is Valid*. This is a filtering rule and is used to verify that the user fulfilling the EWO role is requesting an allowable query. It is checked against the actor's allowable query set, to determine validity.

RuleID 11 is titled *IB Results 2*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of "Vessel Results from Secret IB and Unclassified IB with ID Injection for Oakland Alert." This is done after checking the user's security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from both the Unclassified and Secret level IB (since the user is at the Secret level), and then the required Alert Message insertion from the TS IB. This rule was created to allow for Harbormaster queries from the port of Oakland as an extension to the San Diego port accounted for in our scenario.

RuleID 12 is titled *Negative ID Response to IB for Alert*. This rule provides an acknowledgment of a negative result for the presence of an alert message. This is used to trigger other rules in the ruleset and to ensure that the ID returns a result in all cases. Without this rule, it is possible for the ID to return nothing, which allows for ambiguity in the ruleset. The rule checks if the query requires data insertion, which is sourced from the higher level IBs. If *FALSE*, then the IB that received the query is informed that its response to the user does not need to wait for the ID injection messaging prior to returning results to the user.

RuleID 13 is titled *Oakland Harbormaster query is valid destination 2*. This rule is almost a duplicate of RuleID 4, but is used to support an alternative method of designating roles. In this case, if the role is listed as Harbormaster Oakland, instead of the generic Harbormaster designation, then the rule will process similarly. This is one of the rules used as an extension to our primary scenario, which would allow for the port of Oakland queries to also work in the system. It is used to verify the user role, the role location, the PDP decision, and the type of query before deciding that it is a valid query and should be processed by the IB.

RuleID 14 is titled *PDP Decision 2*. This is an externally sourced rule that would be completed by the Policy Decision Point service in our SOA system. It is used, based on the variables it is sourced with, to provide a decision for that user’s access to resources in the system. In this implementation, it is not referenced from an actual external PDP service, and this rule is intended to provide a positive response to the service check for a negative response.

RuleID 15 is titled *Query is Invalid.* This is a filtering rule and is used to verify that the user (EWO or Harbormaster) is requesting an allowable query. It is sourced from two other rules, which check the validity of the query against each actor’s allowable query set. This rule is intended to provide a positive response to the query check in the case of a negative response.

RuleID 16 is titled *Alert SD Secret IB 2*. This rule is used to verify that when an alert is present at the Secret level IB, intended for San Diego, and the requestor is at the Unclassified security level, that the *Alert_Present_Response* is set to *TRUE.* This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level. This rule is equivalent to RuleID 9 and is used to support an alternative method of designating roles. In this case, if the role is listed as the generic Harbormaster, instead of the Harbormaster San Diego designation, then the rule will process similarly.

RuleID 17 is titled *IB Results 3*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Secret IB and Unclassified IB.” This is done after checking the user’s security level and for the presence of an alert for this location. The IB, in this case, will return its regular result from both the Unclassified and Secret level IB (since the user is at the Secret level).

RuleID 18 is titled *Oakland Harbormaster query is valid origination*. This rule is supports an alternative method of designating roles. In this case, if the role is listed as Harbormaster Oakland, instead of the generic Harbormaster designation, then the rule will process similarly. This is one of the rules used as an extension to our primary scenario, which would allow for the port of Oakland queries to also work in the system. It is used to verify the user role, the role location, the PDP decision, and that the query type is for an Originating Port before deciding that it is a valid query and should be processed by the IB.

RuleID 19 is titled *Alert SD TS IB*. This rule serves as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert Message against the security level of the user producing the query. This rule is used to verify that when an alert is present at the Top Secret level IB, intended for San Diego, the requestor is at the Unclassified or Secret security level, and that the *Alert_Present_Response* is set to *TRUE*. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level.

RuleID 20 is titled *EWO Query is Valid 2*. This is a filtering rule and is used to verify that the user fulfilling the EWO role is requesting an allowable query. It is checked for validity against the actor’s allowable query set. This rule is equivalent to RuleID 10, but accounts for a different role naming convention. In this case, the user's role is Electronic Warfare Officer, instead of EWO.

RuleID 21 is titled *IB Results 4*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with, to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of "Vessel Results from Unclassified IB." This is done after checking the user's security level and for the presence of an alert for this location being *FALSE*. The IB, in this case, would return its regular result from the Unclassified-level IB.

RuleID 22 is titled *Oakland Harbormaster query is valid origination 2*. This rule is almost a duplicate of RuleID 18, but is used to support an alternative method of designating roles. In this case, if the role is listed as the generic Harbormaster, instead of the Harbormaster Oakland designation, then the rule will process similarly. This is one of the rules used as extension to our primary scenario, which would allow for the port of Oakland queries to also work in the system. It is used to verify the user role, the role location, the PDP decision, and the type of query is an originating port query before deciding whether it is a valid query and should be processed by the IB.

RuleID 23 is titled *Alert SD TS IB*. This rule is used to verify that when an alert is present at the Top Secret level IB, intended for San Diego, and the requestor is at the Unclassified or Secret security level, that the *Alert_Present_Response* is set to *TRUE*. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert's classification level. This rule is equivalent to RuleID 19 and is used to support an alternative method of designating roles. In this case, if the role is listed as the generic Harbormaster, instead of the Harbormaster San Diego designation, then the rule will process similarly.

RuleID 24 is titled *IB Results 5*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of "Vessel Results from TS IB, Secret IB and

Unclassified IB.” This is done after checking the user’s security level is equal to *Top Secret*. The user in this case is authorized all data from each of the classification levels and does not require injection to the IB results.

RuleID 25 is titled *SD Harbormaster query is valid destination*. This rule is used to verify the user role as *Harbormaster*, the role location is *San Diego*, the PDP decision is *TRUE*, and the query is both valid and of the type *Destination_Port_Query* before deciding whether it is valid and should be processed by the IB.

RuleID 26 is titled *Alert Oak Secret IB*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert Message against the security level of the user producing the query. This particular rule is used as an extension to our baseline scenario to account for users at the port of Oakland. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level.

RuleID 27 is titled *IB Results 6*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rulebase with a sample query. In this case, it returns the IB result of “Invalid Query!” This is done after checking the query validity from the *Valid_Query* variable.

RuleID 28 is titled *SD Harbormaster query is valid destination 2*. This rule is identical in functionality to RuleID 25, but supports the extended role convention using *Harbormaster San Diego* instead of the generic *Harbormaster* role. This rule is used to verify the user role as *Harbormaster San Diego*, the PDP decision is *TRUE*, and the type of query is *Destination_Port_Query* before deciding that it is valid and should be processed by the IB.

RuleID 29 is titled *Alert Oak Secret IB 2*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert Message against the security level of the user producing the query. It is identical in functionality to RuleID 26, but supports the generic role moniker of *Harbormaster* instead of *Harbormaster Oakland*. This particular rule is used as an extension to our baseline scenario to account for users at the port of Oakland. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert's classification level.

RuleID 30 is titled *SD Harbormaster query is valid origination*. This rule is identical in functionality to RuleID 32, but supports the extended role convention using *Harbormaster San Diego* instead of the generic *Harbormaster* role. This rule is used to verify the user role as *Harbormaster San Diego*, the PDP decision is *TRUE*, and the type of query is *Originating_Port_Query* before deciding that it is valid and should be processed by the IB. This rule checks the query that is being processed and sets conditions to restrict the IB response as a result. This rule is designed to eliminate the user's ability to repeatedly query the system to troll for information, or the lack of information, again a key element in preventing the Misuse Case. In the original system, queries were directly processed by the IB after entry. In the re-architected system, we reroute all queries through the ID (per Figure 26) where we begin the new data flow at continuation point I instead of direct processing by the IB in the original flow. In this case, the query and resultant data must include the Harbormaster's port (either Origination or Destination) for details to be returned by the IB and ID. This is part of the <Detect> in the Security Use Case that we had to reroute queries in the system. This will prevent a Harbormaster from continually querying the system to get information about vessels from the system that do not directly pertain to his or her role (at his location) and his normal performance of duty. This rule verifies that the user is a Harbormaster from San Diego and conducting an Originating Port Query, which he is authorized because of his role's duties and obligations, as well as a PDP service check to validate the role's permissions to execute a basic query and to access information from the IB. The rule then establishes this as a valid query for the San Diego Harbormaster, while it also restricts the query response by limiting Vessel_Classification_Level to the User_Security_Level, and setting the field for Vessel_Originating_Port to the User_Role_Location, or in this case San Diego.

RuleID 31 is titled *Alert Oak TS IB*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert

Message against the security level of the user producing the query. It is identical in functionality to RuleID 33, but supports the generic role moniker of *Harbormaster* instead of *Harbormaster Oakland*. This particular rule is used as an extension to our baseline scenario to account for users at the port of Oakland. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert's classification level.

RuleID 32 is titled *SD Harbormaster query is valid origination 2*. This rule is almost a duplicate of RuleID 30, but is used to support an alternative method of designating roles. In this case, if the role is listed as the generic *Harbormaster*, instead of the *Harbormaster San Diego* designation, then the rule will process similarly. This rule checks the query that is being processed and sets conditions to restrict the IB response as a result. This rule is designed to eliminate the user's ability to repeatedly query the system to troll for information, or the lack of information, serving as a key element in preventing the Misuse Case. In the original system, queries were directly processed by the IB after entry. In the re-architected system, we reroute all queries through the ID (per Figure 26) where we begin the new data flow at continuation point I instead of direct processing by the IB in the original flow. In this case, the query and resultant data must include the Harbormaster's port (either Origination or Destination) for details to be returned by the IB and ID. This is part of the <Detect> in the Security Use Case that we had to reroute queries in the system. This will prevent a Harbormaster from continually querying the system to get information about vessels from the system that do not directly pertain to his or her role (at this location) and their normal performance of duty. This rule verifies that the user is a Harbormaster from San Diego and conducting an Originating Port Query, which he is authorized because of his role's duties and obligations, as well as a PDP service check to validate the role's permissions to execute a basic query and to access information from the IB. The rule then establishes this as a valid query for the San Diego Harbormaster, while it also restricts the query response by limiting Vessel_Classification_Level to the User_Security_Level, and setting the field for Vessel_Originating_Port to the User_Role_Location, or in this case San Diego.

RuleID 33 is titled *Alert Oak TS IB 2*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert Message against the security level of the user producing the query. This rule is identical in functionality to RuleID 30. This particular rule is used as an extension to our baseline scenario to account for users at the port of Oakland. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert's classification level.

RuleID 34 is titled *IB Results 0.1*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of "Vessel Results from Unclassified IB with ID Injection for Oakland Alert." This is done after checking the user's security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from the Unclassified level IB (since the user is at the Unclassified level), and then the required Alert Message insertion from the higher (TS or Secret) IB. The level of the Alert Message is not revealed to the user from the IB. This rule was created to allow for Harbormaster queries from the port of Oakland as an extension to the San Diego port accounted for in our scenario.

RuleID 35 is titled *IB Results 0.2*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with, to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of "Vessel Results from Unclassified IB with ID Injection for San Diego Alert." This is done after checking the user's security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from the Unclassified level IB (since the user is at the Unclassified level), and then the required Alert Message insertion from the higher (TS or Secret) IB. The level of the Alert Message is not revealed to the user from the IB.

RuleID 36 is titled *IB Results 0.3*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with, to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Unclassified IB with ID Injection for San Diego Alert.” This is done after checking the user’s security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from the Unclassified level IB (since the user is at the Unclassified level), and then the required Alert Message insertion from the TS IB. The level of the Alert Message is not revealed to the user from the IB.

RuleID 37 is titled *No Alert IB.* This rule serves as a trigger within the ruleset to identify that an Alert Message is not present at a higher level IB. This indicates a negative search result from within the IB (i.e., no alert message found).

RuleID 38 is titled *IB Results 2.1*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with, to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Secret IB and Unclassified IB with ID Injection for San Diego Alert.” This is done after checking the user’s security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from both the Unclassified and Secret level IB (since the user is at the Secret level), and then the required Alert Message insertion from the TS IB.

RuleID 39 is titled *EWO Query is Invalid.* This is a filtering rule and is used to verify that the user (EWO) is requesting an allowable query. It verifies the role of the query requestor and is intended to provide a positive response to the query check in the case of a negative response (i.e., an invalid query request).

RuleID 40 is titled *HM Query is Invalid.* This is a filtering rule and is used to verify that the user (Harbormaster) is requesting an allowable query. In this case, it allows for Harbormaster queries to originate from San Diego and Oakland only. Any other location for the role of a Harbormaster will make the query invalid. It verifies the role of the query requestor and is intended to provide a positive response to the query check in the case of a negative response (i.e., an invalid query request).

B. REGRESSION TESTING

Several rule engines exist for integration and use with RuleML. For this research we initially selected VDR-Device [68] as a visual integrated development environment and additionally, we used Acumen Business's The Rule Manager [69] to assist with visual rule mapping and stepping through the proof cases for our ruleset with its ability to allow interaction and establish parameter values as the rule trace progresses. As a part of regression testing as each new rule was implemented or changed a full testing process of all rules was conducted again. Three main cases exist for our ruleset to provide a clear and convincing argument that it is sound and complete. The first case is to demonstrate that all intended functionality and usage that was provided for in the original system is still available. This is despite the fact that we are now going through an additional service and utilizing a RuleML ruleset to specify our security policy. The second case is to demonstrate that the unintended use (and the loss of safety) of the system that was described from the misuse case is now prevented. The third and final piece is to show that unintended or ancillary queries are not answered by our system inadvertently. This shows that no additional use is allowed by the implementation of our ruleset than that which was present in the original system.

1. Intended Use Is Maintained

The original functionality is maintained by the system, despite the incorporation of our re-architecting and the implementation of the ID and its ruleset. To show that this still works as it did prior to this, we will process a query using the following parameters:

Type of query:	Destination Port
Role / Actor:	Harbormaster
Location:	San Diego
Security Level:	Unclassified
Alert Present:	False
Alert Classification Level:	Not Applicable
Expected Result:	"Vessel Results from Unclassified IB"

The expected result from this query and the RuleML execution is an IB response of "Vessel Results from Unclassified IB" to the user. The trace of this query execution through the ruleset can be seen in Appendix B, which depicts the query and ruleset using RuleManager's Interactive Rule Map functionality. In Appendix B, the detailed view of the trace also shows the individual variables and terms from within each rule being sourced as the engine completes its evaluation. These terms are referenced by the individual rule and then evaluated to determine the end result for each rule. For this query, Return_IB_Results is the main variable (and expressed at the root of the rule tree) that is sourced from the ten different IB Result rules. Those rules were designed to replicate the external services that were not implemented for this research. Each IB Result rule can begin the trace to evaluate down to the base predicates (terminals in a CFG) that are necessary to resolve or decide the rule. This may be iterative in stepping through other rules to source the necessary values for the rule in question. In the first query instance, the IB Results 2.1 rule (RuleID 38) is the rule chosen to begin the sourcing of Return_IB_Results. The trace is as follows:

1.	RuleID 38–IB Results 2.1	Sourced
2.	RuleID 40–HM Query is Invalid	Fired
3.	RuleID 32–SD Harbormaster query is valid origination 2	Did Not Fire
4.	RuleID 30–SD Harbormaster query is valid origination	Did Not Fire
5.	RuleID 14–Policy Decision Point 2	Did Not Fire
6.	RuleID 5–Policy Decision Point	Fired
7.	RuleID 28–SD Harbormaster query is valid destination 2	Fired
8.	RuleID 25–SD Harbormaster query is valid destination	Did Not Fire
9.	RuleID 22–Oakland Harbormaster query is valid origination 2	Did Not Fire
10.	RuleID 18–Oakland Harbormaster query is valid origination	Did Not Fire
11.	RuleID 13–Oakland Harbormaster query is valid destination 2	Did Not Fire

12. RuleID 4–Oakland Harbormaster query is valid destination	Did Not Fire
13. RuleID 39–EWO query is invalid	Fired
14. RuleID 20–EWO query is valid 2	Did Not Fire
15. RuleID 10–EWO query is valid	Did Not Fire
16. RuleID 15–Query Invalid	Fired
17. RuleID 7–Query is Valid	Fired
18. RuleID 37–No Alert IB	Fired
19. RuleID 33–Alert Oak TS IB 2	Did Not Fire
20. RuleID 31–Alert Oak TS IB	Did Not Fire
21. RuleID 29–Alert Oak Secret IB 2	Did Not Fire
22. RuleID 26–Alert Oak Secret IB	Did Not Fire
23. RuleID 23–Alert SD TS IB 2	Did Not Fire
24. RuleID 19–Alert SD TS IB	Did Not Fire
25. RuleID 16–Alert SD Secret IB 2	Did Not Fire
26. RuleID 9–Alert SD Secret IB	Did Not Fire
27. RuleID 0–Alert Not Valid for Role Location	Fired
28. RuleID 8–Alert Notification is Absent	Fired
29. RuleID 1–Alert Notification is Present	Did Not Fire
30. RuleID 12–Negative ID Response to IB for Alert	Fired
31. RuleID 6–Positive ID Response to IB for Alert	Did Not Fire
32. RuleID 38–IB Results 2.1	Did Not Fire
33. RuleID 36–IB Results 0.3	Did Not Fire
34. RuleID 35–IB Results 0.2	Did Not Fire
35. RuleID 34–IB Results 0.1	Did Not Fire

36. RuleID 27–IB Results 6	Did Not Fire
37. RuleID 24–IB Results 5	Did Not Fire
38. RuleID 21–IB Results 4	Fired
39. RuleID 17–IB Results 3	Did Not Fire
40. RuleID 11–IB Results 2	Did Not Fire
41. RuleID 3–IB Results 0	Did Not Fire

RuleIDs 0–40 were each evaluated and either fired or did not fire. The originating rule (Step 1. IB Results 2.1) to begin the sourcing *Did Not Fire* in Step 32. Despite the ruleset finding its result after Step 38 when RuleID 21 Fired, the remaining rules are still evaluated in the ruleset for completeness. The RuleID 21 for IB Results 4 produced the expected result for this query of “Vessel Results from Unclassified IB.” This query effectively shows that the ruleset does not hinder the intended functionality or usage of the system. All original queries that were allowed are still supported with no change to the resultant data.

2. Unintended Use Is Prevented

The second case for our clear and convincing argument consists of our safety property violation for information flow. The misuse of the system must be prevented by our re-architecting and the implementation of the ID and its ruleset. To show that this misuse has been prevented we will process a query using the following parameters:

Type of query:	Destination Port
Role / Actor:	Harbormaster
Location:	San Diego
Security Level:	Unclassified
Alert Present:	True
Alert Classification Level:	Secret
Expected Result:	“Vessel Results from Unclassified IB with ID Injection for Port of San Diego”

The expected result from this query and the RuleML execution is an IB response of “Vessel Results from Unclassified IB with ID Injection for Port of San Diego” to the user. The trace of this query execution through the ruleset can be seen in Appendix C, which depicts the query and ruleset using RuleManager’s Interactive Rule Map functionality. In Appendix C, the detailed view of the trace also shows the individual variables and terms from within each rule being sourced as the engine completes its evaluation. These terms are referenced by the individual rule and then evaluated to determine the end result for each rule. For this query, Return_IB_Results is the main variable (and expressed at the root of the rule tree) that is sourced from the ten different IB Result rules. Those rules were designed to replicate the external services that were not implemented for this research. Each IB Result rule can begin the trace to evaluate down to the base predicates (terminals in a CFG) that are necessary to resolve or decide the rule. This may be iterative in stepping through other rules to source the necessary values for the rule in question. In the first query instance, the IB Results 2.1 rule (RuleID 38) is the rule chosen to begin the sourcing of Return_IB_Results. The trace is as follows:

1.	RuleID 38–IB Results 2.1	Sourced
2.	RuleID 40–HM Query is Invalid	Fired
3.	RuleID 32–SD Harbormaster query is valid origination 2	Did Not Fire
4.	RuleID 30–SD Harbormaster query is valid origination	Did Not Fire
5.	RuleID 14–Policy Decision Point 2	Did Not Fire
6.	RuleID 5–Policy Decision Point	Fired
7.	RuleID 28–SD Harbormaster query is valid destination 2	Fired
8.	RuleID 25–SD Harbormaster query is valid destination	Did Not Fire
9.	RuleID 22–Oakland Harbormaster query is valid origination 2	Did Not Fire
10.	RuleID 18–Oakland Harbormaster query is valid origination	Did Not Fire
11.	RuleID 13–Oakland Harbormaster query is valid destination 2	Did Not Fire
12.	RuleID 4–Oakland Harbormaster query is valid destination	Did Not Fire
13.	RuleID 39–EWO query is invalid	Fired

14. RuleID 20–EWO query is valid 2	Did Not Fire
15. RuleID 10–EWO query is valid	Did Not Fire
16. RuleID 15–Query Invalid	Fired
17. RuleID 7–Query is Valid	Fired
18. RuleID 37–No Alert IB	Did Not Fire
19. RuleID 33–Alert Oak TS IB 2	Did Not Fire
20. RuleID 31–Alert Oak TS IB	Did Not Fire
21. RuleID 29–Alert Oak Secret IB 2	Did Not Fire
22. RuleID 26–Alert Oak Secret IB	Did Not Fire
23. RuleID 23–Alert SD TS IB 2	Did Not Fire
24. RuleID 19–Alert SD TS IB	Did Not Fire
25. RuleID 16–Alert SD Secret IB 2	Did Not Fire
26. RuleID 9–Alert SD Secret IB	Fired
27. RuleID 0–Alert Not Valid for Role Location	Did Not Fire
28. RuleID 8–Alert Notification is Absent	Did Not Fire
29. RuleID 1–Alert Notification is Present	Fired
30. RuleID 12–Negative ID Response to IB for Alert	Did Not Fire
31. RuleID 6–Positive ID Response to IB for Alert	Fired
32. RuleID 38–IB Results 2.1	Did Not Fire
33. RuleID 36–IB Results 0.3	Did Not Fire
34. RuleID 35–IB Results 0.2	Fired
35. RuleID 34–IB Results 0.1	Did Not Fire
36. RuleID 27–IB Results 6	Did Not Fire
37. RuleID 24–IB Results 5	Did Not Fire

38. RuleID 21–IB Results 4	Did Not Fire
39. RuleID 17–IB Results 3	Did Not Fire
40. RuleID 11–IB Results 2	Did Not Fire
41. RuleID 3–IB Results 0	Did Not Fire

RuleIDs 0–40 were each evaluated and either fired or not fired. The originating rule (Step 1. IB Results 2.1) to begin the sourcing *Did Not Fire* in Step 32. Despite the ruleset finding its result after Step 34 when RuleID 35 Fired, the remaining rules are still evaluated in the ruleset for completeness. The RuleID 35 for IB Results 0.2 produced the expected result for this query of “Vessel Results from Unclassified IB with ID Injection for Port of San Diego.” This query effectively shows that the ruleset does intervene and provide for injection of data to prevent the misuse case. This eliminates the ability of the Harbormaster to infer information from the system and regains the safety of our information flows.

3. Additional Use Is Excluded

The original functionality is maintained by the system, despite the incorporation of our re-architecting and the implementation of the ID and its ruleset. To show that this still works as it did prior to this, we will process a query using the following parameters:

Type of query:	Destination Port
Role / Actor:	Harbormaster
Location:	San Diego
Security Level:	Unclassified
Alert Present:	False
Alert Classification Level:	Not Applicable
Expected Result:	“Invalid Query!”

The expected result from this query and the RuleML execution is an IB response of “Invalid Query!” to the user. The trace of this query execution through the ruleset can be seen in Appendix D, which depicts the query and ruleset using RuleManager’s Interactive Rule Map functionality. In Appendix D, the detailed view of the trace also shows the individual variables and terms from within each rule being sourced as the engine completes its evaluation. These terms are referenced by the individual rule and then evaluated to determine the end result for each rule. For this query, Return_IB_Results is the main variable (and expressed at the root of the rule tree) that is sourced from the ten different IB Result rules. Those rules were designed to replicate the external services that were not implemented for this research. Each IB Result rule can begin the trace to evaluate down to the base predicates (terminals in a CFG) that are necessary to resolve or decide the rule. This may be iterative in stepping through other rules to source the necessary values for the rule in question. In the first query instance, the IB Results 2.1 rule (RuleID 38) is the rule chosen to begin the sourcing of Return_IB_Results. The trace is as follows:

1. RuleID 38–IB Results 2.1	Sourced
2. RuleID 40–HM Query is Invalid	Fired
3. RuleID 32–SD Harbormaster query is valid origination 2	Did Not Fire
4. RuleID 30–SD Harbormaster query is valid origination	Did Not Fire
5. RuleID 28–SD Harbormaster query is valid destination 2	Did Not Fire
6. RuleID 25–SD Harbormaster query is valid destination	Did Not Fire
7. RuleID 22–Oakland Harbormaster query is valid origination 2	Did Not Fire
8. RuleID 18–Oakland Harbormaster query is valid origination	Did Not Fire
9. RuleID 13–Oakland Harbormaster query is valid destination 2	Did Not Fire
10. RuleID 4–Oakland Harbormaster query is valid destination	Did Not Fire
11. RuleID 39–EWO query is invalid	Fired
12. RuleID 20–EWO query is valid 2	Did Not Fire
13. RuleID 10–EWO query is valid	Did Not Fire
14. RuleID 15–Query Invalid	Fired
15. RuleID 7–Query is Valid	Did Not Fire
16. RuleID 38–IB Results 2.1	Did Not Fire
17. RuleID 36–IB Results 0.3	Did Not Fire

18. RuleID 35–IB Results 0.2	Did Not Fire
19. RuleID 34–IB Results 0.1	Did Not Fire
20. RuleID 27–IB Results 6	Fired
21. RuleID 24–IB Results 5	Did Not Fire
22. RuleID 21–IB Results 4	Did Not Fire
23. RuleID 17–IB Results 3	Did Not Fire
24. RuleID 11–IB Results 2	Did Not Fire
25. RuleID 3–IB Results 0	Did Not Fire

RuleIDs 0–40 were not all evaluated for this query. The resultant value for the IB did not require information or variables from part of the ruleset in this query. Because we did not allow for actors outside of the scope of our scenario, the query processed was filtered by the ID and deemed *Invalid*. The originating rule (Step 1. IB Results 2.1) to begin the sourcing *Did Not Fire* in Step 16. Despite the ruleset finding its result after Step 20 when RuleID 27 Fired, the remaining rules for IB Results are still evaluated in the ruleset for completeness because they contain the same sourcing predicates that are used by the other IB Result rules and map to the final *Return_IB_Results* variable. The RuleID 27 for IB Results 6 produced the expected result for this query of “Invalid Query!” This query and the trace of our RuleML ID ruleset provide a clear and convincing argument that the ruleset does not support unintended usage or additional functionality that would not be supported by the original implementation.

C. SUMMARY

In order to support the goal of information sharing, we must not only consider information releasability, but non-disclosure of other information. The flexibility that is offered through the Information Declassifier and its execution of our rule base for all Information Broker interactions provides a positive step toward our ultimate goal of secure information sharing. We have clearly shown through these three test cases the viability of using RuleML as a CDS information flow control specification language and particularly within a SOA-based environment. The flexibility of RuleML to account for the de-classification problem, resultant from the BLP tranquility principle, is limited only by the developmental ability of the human drivers creating the underlying policy in RuleML.

VII. CONCLUSION

A. CONTRIBUTIONS

In this research, we have demonstrated that by introducing the contributions listed below the safety properties of a mixed model access control, SOA-based, MLS system can be preserved, despite emergent challenges resultant from the composition of those disparate access controllers in order to facilitate information sharing. Table 12 provides a summary of the contributions of the research presented in this dissertation.

The specific contributions of this work are:

1. Developed and Used a New Methodology for Security Analysis using UML-based Use Case Analysis to Direct the Re-architecting of a System for the Preservation of Safety

The methodology of security analysis using UML Use and Misuse Cases, allows for tailoring of security policies and mitigating the "highest cost" risks for information flow while still enabling trusted sharing. UML-based Use Case security analysis has not been applied in a MLS, SOA-based venue and this work utilizes that approach as an exemplar to provide for a greater access to information and increased sharing among disparate security domains. In order to support inviolate information flows to the security policy, and to overcome the tranquility property associated with traditional MLS systems, we demonstrate that Radiant Alloy needs to use an information declassifier.

Contribution	❖ Provide a method for security analysis using UML Use Case Analysis to direct the re-architecting of a system for the preservation of safety. ❖ Incorporate a process to allow for the prioritization of information flows and the "safe" composition of mixed access control models.	❖ Create business process rules, using RuleML to specify allowed information flows and to restrict flows that enable leakage of information from emergent behaviors, and facilitate sharing between systems.	❖ Validation of the system re-architecting through the conduct of a Regression Test to provide verification of RuleML content based query injection and filtering.
Impact to DOD	❖ Allows for tailoring security policies and mitigating the "highest cost" risks for information flow while still enabling trusted sharing. ❖ Opens the information sharing infrastructure to more DOD organizations.	❖ Allow greater control of Need to Share / Need Not to Share information within domains. ❖ Increase speed of information dissemination. ❖ Address high tempo of battle and change in situation for MDA instances.	❖ Build basis of confidence for services use/reuse with respect to the Information Broker and its associated rule engine. ❖ Build basis of confidence for services use/reuse with respect to the Information Broker and its associated rule engine.
Impact to Software Engineering	❖ Utilizes a process in a new domain and to provide for a greater granularity of access to information. ❖ Allows engineers and developers to combine "best of" practices in their design of an information system.	❖ Extends an open-source, XML-based, business process language to facilitating the composition of access controllers and the safety of the associated models.	❖ Automatically provide Information Broker web services a means of information sharing and information filtering that is not available within individual access control models.

Table 12. Research Contribution Summary

2. Developed a New Process to Allow for the Prioritization of Information Flows and the "Safe" Composition of Mixed Access Control Models through a Re-architecting of the System

Through the process of prioritizing the information-flow needs within a system, we can tailor the composition of different access control models to support required flows. This effectively opens the information sharing infrastructure to more DOD organizations and provides a means for cooperation and more real-time passing of information between agencies. This process also helps engineers and developers combine "best-of" practices in their design and implementation of an AIS. The idea of prioritization is based on the risk analysis and risk tolerances of the stakeholders. The allowable "exceptions" to our baseline policies are considered as the only permissible extensions and ultimately determine what is safe for the system. The method shown in this research uses RuleML to enforce this policy. The need for sharing and the need to maintain security must be balanced to provide an operationally useful system that will still uphold the desired and specified safety for information flows.

3. Created a New Process for the Development of Business Process Rules, using RuleML to Specify Allowed Information Flows and to Restrict Flows Enabling Leakage of Information from Emergent Behaviors and Facilitate Sharing between Information Systems

To support the development of an information declassifier, we provide a policy-based framework to do so using RuleML [15]. By developing, implementing and enforcing business process rules through the use of RuleML, we realize a greater control of our information flows. The rule engine provides for a greater granularity of Need (Not) to Share for information within domains, facilitating the dissemination of information and providing increased speed of information flow to allow a greater use of "real-time" information, compared with traditional MAC-only policies. Current systems cannot support this level of information sharing and traditionally use a workaround (i.e., sneakernet) to meet user requirements. The extension of an open-source XML-based business process language to facilitate the composition of access controllers and the safety of the associated models provides a useful tool for software engineers in system-architecting efforts.

4. Provided a Process that Allows Engineers to use Decision Tables or Other Methods to Develop Simple Propositions to Express Desired Process and Information Flows that can be Formed into Executable RuleML

With the re-architecting of an information system, it is necessary to provide a convincing argument that the RuleML specification supports the system's required information flows. By creating business process rules with RuleML to support the sharing of information within the system, we can show that all prior capabilities and intended flows are still available to the user, yet there are no unintended flows that result in violation of the safety property. This is an extension of the work conducted on the hook-up theorem for multilevel security with respect to inference control and the composability of restrictiveness for security policies [41]. This must be done to build a basis of confidence for services to be used (and reused) with respect to the Information Broker and its associated rule engine based Information Declassifier service. By regression testing, which includes running the ruleset with a sample data set, we can verify that our security use case is correct. This is necessary to help establish a basis for certification of information systems at high assurance levels. The rule-based, Information Declassifier service helps to automatically provide Information Broker web services with a means of information sharing and information filtering that is not available within individual access control models, and to ensure the safety of information flows in mixed model access control systems. This included the validation of the concept system re-architecting through the conduct of a regression testing verification and a simulation of RuleML content-based query injection and filtering, whose results verified the rule structure and usage of RuleML as a specification language.

B. FUTURE WORK

Research and engineering development remains to be done to provide for the seamless integration of disparate domain, mixed model access control information systems with high assurance information sharing as we envision. RuleML provides a technically feasible way of specifying cross-domain information flow control policies and for implementing an information declassifier, but work remains to be done to fully explore how to leverage the power of RuleML. In addition, there are many specialty areas that attempt to use cross-domain solutions for integration of repositories and information sharing. While each of these potential areas offers great potential, much of the realization involves the willingness to partner and share between disparate organizations, and hence the solution to the problem cannot be purely technical.

1. Malware and Advanced Persistent Threat Correlation

Malware has become a pervasive element in computing today and a key element in gaining access to a protected network. The hardest threat to defend becomes the advanced persistent threat (APT), which is typically state-sponsored. This APT is resourced, well-trained, and active in state-sponsored entities. The effort to defend an enterprise network against these threats and the services being offered to assist in this process are continually growing. The resources required to defend against malware and hackers in terms of monetary and overall resource cost, time, and scope of effort have increased dramatically each year and the expertise of the APT heightens this effort even more. The effort is asymmetrical, where the defender has to determine what to defend and how to defend against an infinite (in theory) number of possible attacks, but the attacker only has to find one or a few exploitable vulnerabilities in order to exploit the network security. While the government and private industry have gone in different directions with tracking and reporting efforts, the defense techniques, the threat activity, and threat actors are the same. Several challenges exist, ranging from political, legal, and policy (e.g., antitrust laws) that must be overcome before technical solutions can be applied. By enabling information sharing and integration of disparate tracking efforts, we could realize an economy of scale in these defense efforts, as well as provide much better intelligence against these known threats.

The Department of Defense (DOD) and Department of Homeland Security (DHS) established a voluntary sharing program for cyber incidents with many industry partners. This is referred to as the Defense Information Base (DIB) and consists of companies such as Northrop Grumman, Raytheon, and Boeing. These DIB participant companies have a strong partnership for threat detection and incident sharing among themselves, but they are only given limited scope information from government sources like the Federal Bureau of Investigation (FBI), and the National Security Agency (NSA) that affords them contextual insight within their own monitoring and network defense practices. However, this effort does not allow for disclosure of key elements that would save time, effort, and financial resources among the partners. Additionally, these same partners are conducting reverse malware analysis and reporting on the details of the threats and malware encountered within their networks, yet the DOD and DHS systems and policies do not currently support that level of bi-directional sharing. By reducing the scope and limiting the details, these DIB partners are being subject to and penetrated by cyber threats that could be avoided with additional sharing. Much of the analysis of the threats is redacted, as are the pseudonyms used between government and DIB partners. This is the primary cause of duplication of effort and a much slower response and reaction time to realized threats for U.S. national security.

By enacting a trusted sharing system with an information broker and an information declassifier as described in this research, this partnership could be greatly enhanced without risk of exposing need-not-to-share type information across non-government partners and between DIB member companies. The research opportunity for synchronizing the disparate tracking repositories for known malware and how that could be better shared through trusted connectors and an established RuleML-based security policy could have a significant effect on the DOD and private industry.

2. Distribution of Services and Rule Sourcing

Our research considers a localized RuleML ruleset. This ruleset was not established as a service, but only tested for viability as a potential information flow policy specification language for the implementation of the Information Declassifier. One of the most needed areas to be covered as an extension of this work is for a full-scale implementation. This implementation should include distributed services and distributed rule sourcing. The work should replicate the services of a SOA system and have rules access different services for firing and decisions. Our work replicated the sourcing of user attributes during rule evaluation and execution, yet this is an area of great concern for a

MLS ID. This would be useful in a distributed environment and give a viable argument for information releasability being safe. The full-scale system should include a distribution and versioning of the de-classification policies in which every agent runs its own policy and calls the other one for the appropriate invocation of rules. This way each agent only exposes those aspects of the policy that they are permitted to expose. One risk associated with this distribution and remote sourcing of rules is with failure and divergence. When we begin to rely on externally sourced services to provide input for rule-evaluation, we risk that the service will hang while waiting for a response. This response may be delayed somewhere in the network, or even worse, the response may be a circular reference and be dependent on the same service that had invoked it for part of its evaluation. This dependency on variables sourced in other rules or other services for successive calls is an area that also needs to be addressed as a function of the distribution of rule sourcing and the full implementation of services. The replication of the ID needs to be done in tandem with sharing the ruleset that is created between IB domains. Since the IB is replicated and the ID will be replicated, an issue can arise in which the ruleset becomes out of synch and the rules from a higher level are not the same as the rules from the lower, since we need to evaluate different risks at disparate domain levels. Additionally, the vessel information that is processed could be sourced from a working U.S. Navy system like MASTER. This system integrates vessel data with the MIEM and could easily populate a query with real data. The MIEM schemas can also allow for a finer granularity of rule based on other viable vessel data fields. Ultimately, the distributed system version would play favorably toward a true and expressed need in the DOD and Homeland Defense communities [5].

3. Using RuleML for a CDS Data Sanitization Policy

There are many venues where this effort can help to realize the information sharing needs of multiple organizations. In particular, we often need to focus on the releasability of information and not just injection or filtering. One of the areas where this research work can be extended is data sanitization between domains. This methodology and development of rules can be tailored to meet the business process needs in just such a manner. Oftentimes federal agencies are tasked to work with state and local agencies to conduct drug enforcement or other special operations. Such scenarios provide fertile ground for exploring information releasability and the use of a sanitizer. Usually when working with the DOD, a state and local agency will provide information to the DOD. By our current policies and classification marking system, as soon as the DOD agency obtains that information, it becomes classified at a higher level than what we can release back to the state or local agency. This is similar to the scenario we worked with in our research, yet the primary reason the DOD is involved is to support the state and local agencies and not the other way around. In this case, we may be able to sanitize certain fields (using XML tags) of data that DOD cannot release back to the state or local agency. But, if we look at the origination of the data and the mission scenario, we should be able to tailor our ruleset accordingly while preserving our ability to enforce our security policy. An additional area may be the consideration of time in the release of information. When an action has already occurred, what value does the information have? Adding a temporal constraint or aspect to a ruleset would help to enforce the time value of information and to enhance our ability to share without violation of policy. This would raise the complexity of analyzing the data itself for queries and rule execution, but would also help to alleviate the inevitable progression of data to the high side of our classification systems.

4. Formal Patterns for Access Control Model Composition

A key area of possible work exists with the integration of disparate access control models. This integration would be served best through the development of a pattern for how a set of access control models can be combined. Software engineering has been the benefactor of both design and architectural patterns for years in the creation of new systems. Patterns trace their origins back to Alexander’s work on urban planning and architecture, and software patterns in particular were brought to prominence in 1994, with the Gamma, Helm, Johnson, and Vlissides’s (the Gang of Four) book of Design Patterns, which was geared toward object oriented programming. [70] Patterns are a best practice approach to solving a recurring problem in a particular domain or context. Design patterns are a generalized approach to solving a problem that must be implemented each time they are used. These patterns are used to address a specific element of the system design. In [70], four essential elements of a design pattern include:

1. A meaningful name
2. A description of the problem to show how and when the pattern should be applied
3. A solution description of the parts of the design solution, relationships, and responsibilities.
4. A statement of consequences including the results and the trade-offs of using or applying the pattern.

This standard pattern format, to which we generally adhere, helps to explain and document the purpose, intent, consequences, and other referencing patterns. Each pattern discussion includes; intent, motivation, applicability, structure, participants, collaborations, consequences, implementation, sample code, known uses and related patterns. Patterns can be used to describe a system structure that meets the design needs of an application in a given domain. Almost as a tribal knowledge is passed along, the pattern use and standardized format allow for an easy to use and robust set of information about rationale, consequences, and related decisions for system design. The pattern also allows for a faster method of documentation, since many of the aspects of "good" documentation are included in the original pattern description. This type of documentation for the composition of disparate access controllers could be of tremendous benefit to the security personnel tasked with integrating multiple systems.

A key benefit to pattern use is the ability to tailor the purpose of the pattern to meet differing quality aspects or critical requirements, such as: performance, security, safety, availability, reliability, maintainability, or dependability. Inherent with this focused approach to maximizing or targeting one aspect is that the others are, of course, affected. This affect may be positive (e.g., security is patterned, but safety is also enhanced) or negative (e.g., security is patterned, but performance is degraded). By utilizing the idea of patterns in our composition of access control models we can afford system and security policy designers the ability to rapidly create security policies that will work well together and make transparent the engineering tradeoff decisions that need to be made. The creation of an ID service and its associated ruleset using a pattern to seed the development is ideal. The composition pattern should probably include areas where the access control models conflict between themselves, the overall information flows and recommendations for how those conflicts can and should be resolved. Any ambiguity in a ruleset is bad. The pattern idea extended to the ID ruleset can allow for conflict-of-interest mitigation (i.e., using rules to decide between conflicts) between systems using differing access control models that are required to share information seamlessly. If we continue to rely on the best efforts of those who are designing a system and trying to integrate disparate policies without such a template, then we risk that policy being unsound or incomplete.

This patterning would also allow for a repeatable methodology for misuse/breach of the safety property between disparate controllers. Ultimately, we face a compromise between the effectiveness of our security mechanisms and controls and the operational effectiveness of the system. A pattern approach to provide a baseline for the mandated policy specified with RuleML (for the Information Declassifier) would help to meet our need for information sharing.

5. RBAC for Mandatory Access Control of Query Sets

Developing RBAC-based permissions for queries to the system and even rules based upon roles for within a specific domain is a promising area for further research. By using role-based permissions on a set of queries (i.e., a single query or a group of *n* queries) assigned to a particular role. This would, in effect, limit the scope of queries by what role the user plays in the system and also add a greater granularity with the access control assisting the information declassifier to fulfill its duties. This would leverage the usefulness of RBAC, where we do not need to change the role-permission mappings very often. Instead, we merely change the user-role mappings for our system. If we could map a set of queries as just another object in the system, then the role could be given specific access to that object (i.e., a restricted set of queries). Then the Information Declassifier that we have described in this work would still be used to filter and inject information between domains. For example, a Harbormaster would be given query ability for two to three queries and RuleML could be used to further filter those role-based queries for location (e.g., the role-based query allows for queries about ships inbound and outbound for any harbormaster to use, but the San Diego HM only gets results for San Diego as filtered by the ID). Additionally, we could utilize web services to invoke separate IDs based on the role-permission mapping by changing the port and addressing of the individual queries based on the role the user is performing. This would allow for a more structured filtering ability, yet it would add to the complexity of designing numerous rulesets for multiple ID services at the same classification-domain level.

6. Automatic Generation of Cover Stories

The use of obfuscation through polyinstantiation to prevent inference attacks and general inference from authorized use is well known. Another area where a RuleML application of policy could be implemented is with the automatic generation of cover stories to promote data hiding. By allowing the automatic generation the system will not continue to produce the same consistent result from a query or interaction. This will also prevent the ability to infer information from the system from both the lack of information presented and from the presence of known invalid information.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. RULESET EXPLICATION

RuleID 0 is titled *Alert Not Valid for Role Location*. This rule is intended to eliminate possibilities for Alert Messaging responses that are not valid for our scenario. If an Alert exists for Los Angeles in the system (at a higher level IB), then this rule turns that notification to *FALSE*, since it does not apply to the port of San Diego.

```
<Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert Not Valid for Role Location</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/10/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule0</oid>
      <!--Alert Not Valid for Role Location-->
      <if>
        <Or>
          <Or>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
```

```
         type=“string”>“Alert Exists (SECRET IB) for Port of
Los Angeles”</Ind>
       </rhs>
      </Equal>
      <Equal>
       <lhs>
        <Atom>
         <Rel>Alert_Notification</Rel>
         <Var>Subject</Var>
        </Atom>
       </lhs>
       <rhs>
        <Ind
         type=“string”>“Alert Exists (TS IB) for Port of Los
Angeles”</Ind>
       </rhs>
      </Equal>
     </Or>
     <Equal>
      <lhs>
       <Atom>
        <Rel>Alert_Notification</Rel>
        <Var>Subject</Var>
       </Atom>
      </lhs>
      <rhs>
       <Ind
        type=“string”>“No Alert Exists”</Ind>
      </rhs>
     </Equal>
    </Or>
   </if>
   <do>
    <Atom>
     <Rel>Alert_Present_Response</Rel>
     <Var>Subject</Var>
     <Ind
      type=“bool”>false</Ind>
    </Atom>
   </do>
  </Rule>
```

Listing 11. RuleML code for RuleID 0

RuleID 1 is titled *Alert Notification is Present*. This rule is intended to set the variable *Query_Requires_Data_Insertion* to *TRUE*, if the higher level IB possesses an Alert Message. In this ruleset, the external reference is not made, but invoked through the use of an internally generated variable for the sourcing.

```
  <Rule
   style=“active”
   evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>Alert Notification is Present</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>2/26/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule1</oid>
  <!--Alert Notification is Present-->
  <if>
    <Equal>
      <lhs>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
        </Atom>
      </lhs>
      <rhs>
        <Ind
          type=“bool”>true</Ind>
      </rhs>
    </Equal>
  </if>
  <do>
    <Atom>
      <Rel>Query_Requires_Data_Insertion</Rel>
      <Var>Subject</Var>
      <Ind
        type=“bool”>true</Ind>
    </Atom>
  </do>
</Rule>
```

Listing 12. RuleML code for RuleID 1

RuleID 2 is titled *Est Location for Role*. This rule is intended to fix the assignment of the user’s role location to the results being returned from the IB. This will allow for additional filtering to match the user’s location with vessel results from the data repositories that match with origination or destination ports.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Est Location for Role</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule2</oid>
      <!--Est Location for Role-->
      <if>
        <Or>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Oakland”</Ind>
            </rhs>
          </Equal>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“San Diego”</Ind>
            </rhs>
          </Equal>
        </Or>
      </if>
      <do>
        <Atom>
          <Rel>User_Role_Location</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>Location</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
      </do>
    </Rule>
```

Listing 13. RuleML code for RuleID 2

RuleID 3 is titled *IB Results 0*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Unclassified IB with ID Injection for Oakland Alert.” This is done after checking the user’s security level and for the presence of an alert for this location. This rule was created to allow for Harbormaster queries from the port of Oakland as an extension to the San Diego port accounted for in our scenario.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>IB Results 0</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
```

```
       <Ind>Randy Arvay</Ind>
      </Fun>
     </Expr>
     <Expr>
      <Fun
       uri="dc:date">
       <Ind>3/5/2009</Ind>
      </Fun>
     </Expr>
    </Plex>
   </label>
   <scope>
    <Ind
     uri="#Information_Declassifier" />
   </scope>
   <oid>rule3</oid>
   <!--IB Results 0-->
   <if>
    <And>
     <And>
      <And>
       <Atom>
        <Rel>Valid_Query</Rel>
        <Var>Subject</Var>
       </Atom>
       <Atom>
        <Rel>IB_Response_Wait</Rel>
        <Var>Subject</Var>
       </Atom>
      </And>
      <Equal>
       <lhs>
        <Atom>
         <Rel>User_Security_Level</Rel>
         <Var>Subject</Var>
        </Atom>
       </lhs>
       <rhs>
        <Ind
         type="string">"Unclassified"</Ind>
       </rhs>
      </Equal>
     </And>
     <Equal>
      <lhs>
       <Atom>
        <Rel>Alert_Notification</Rel>
        <Var>Subject</Var>
       </Atom>
      </lhs>
      <rhs>
       <Ind
        type="string">"Alert Exists (TS IB) for Port of
Oakland"</Ind>
      </rhs>
```

```
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Unclassified IB with ID
Injection for Oakland Alert”</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 14. RuleML code for RuleID 3

RuleID 4 is titled *Oakland Harbormaster query is valid destination*. This is one of the rules used as extension to our primary scenario, which would allow for the port of Oakland queries to also work in the system. It is used to verify the user role, the role location, the PDP decision, and the type of query before deciding that it is a valid query and should be processed by the IB.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Oakland Harbormaster query is valid
destination</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule4</oid>
      <!--Oakland Harbormaster query is valid destination-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Harbormaster”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role_Location</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Oakland”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>HM_Queries</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Destination_Port Query”</Ind>
              </rhs>
            </Equal>
          </And>
          <Atom>
            <Rel>PDP_Decision</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
```

```
      <Var>Subject</Var>
      <Ind
        type=“bool”>true</Ind>
    </Atom>
    <Atom>
      <Rel>Vessel_Classification_Level</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
    <Atom>
      <Rel>Vessel_Destination_Port</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Role_Location</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
  </do>
</Rule>
```

Listing 15. RuleML code for RuleID 4

RuleID 5 is titled *PDP Decision*. This is an externally sourced rule that would be completed by the Policy Decision Point service in our SOA system. It is used, based on the variables it is sourced with, to provide a decision for that user’s access to resources in the system. In this implementation, it is not referenced from an actual external PDP service.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>Policy Decision Point</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
```

```
              uri=“dc:date”>
              <Ind>2/26/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule5</oid>
      <!--Policy Decision Point-->
      <if>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Role_Permissions</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“bool”>true</Ind>
          </rhs>
        </Equal>
      </if>
      <do>
        <Atom>
          <Rel>PDP_Decision</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 16. RuleML code for RuleID 5

RuleID 6 is titled *Positive ID Response to IB for Alert*. This rule is used to indicate a positive response to the presence of an Alert Message at a higher level IB and verifies that the query to the IB is valid based on the roles active in our system. This rule ensures that the IB response that was previously returned directly to the user is now processed through the ID and missing data is injected to the query if necessary, to eliminate the Misuse Case and as a <Prevent> feature of our Security Use Case. This rule is designed to verify other rules that determine if the query to the system is valid, and if the response from the higher level IB requires the system to inject data before processing the query response. It also checks if the query requires data insertion, which is sourced from the higher level IBs. If both of these conditions are *TRUE*, then the IB that received the query is informed that its response to the user must wait for the injection messaging from the ID service prior to returning results to the user.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Positive ID Response to IB for Alert</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>12/5/2008</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule6</oid>
      <!--Positive ID Response to IB for Alert-->
      <if>
        <And>
          <Atom>
            <Rel>Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
          <Atom>
            <Rel>Query_Requires_Data_Insertion</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>IB_Response_Wait</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
```

```
       </Atom>
     </do>
   </Rule>
```

Listing 17. RuleML code for RuleID 6

RuleID 7 is titled *Query is Valid*. This is a filtering rule and is used to verify that the user (EWO or Harbormaster) is requesting an allowable query. It is sourced from two other rules, which check if the query against each actor's allowable query set, to determine validity.

```
  <Rule
     style=“active”
     evaluation=“strong”>
     <label>
       <Plex>
         <Expr>
           <Fun
             uri=“dc:title”>
             <Ind>Query is Valid</Ind>
           </Fun>
         </Expr>
         <Expr>
           <Fun
             uri=“dc:author”>
             <Ind>Randy Arvay</Ind>
           </Fun>
         </Expr>
         <Expr>
           <Fun
             uri=“dc:date”>
             <Ind>3/10/2009</Ind>
           </Fun>
         </Expr>
       </Plex>
     </label>
     <scope>
       <Ind
         uri=“#Information_Declassifier” />
     </scope>
     <oid>rule7</oid>
     <!--Query is Valid-->
     <if>
       <Or>
         <Atom>
           <Rel>HM_Valid_Query</Rel>
           <Var>Subject</Var>
         </Atom>
         <Atom>
           <Rel>EWO_Valid_Query</Rel>
           <Var>Subject</Var>
         </Atom>
       </Or>
     </if>
     <do>
       <Atom>
         <Rel>Valid_Query</Rel>
         <Var>Subject</Var>
         <Ind
           type=“bool”>true</Ind>
       </Atom>
     </do>
   </Rule>
```

Listing 18. RuleML code for RuleID 7

RuleID 8 is titled *Alert Notification is Absent*. This rule is intended to set the variable *Query_Requires_Data_Insertion* to *FALSE*, if the higher level IB does not possess an Alert Message. In this ruleset, the external reference is not made, but invoked through the use of an internally generated variable for the sourcing.

```
<Rule
   style=“active”
   evaluation=“strong”>
   <label>
     <Plex>
       <Expr>
         <Fun
           uri=“dc:title”>
           <Ind>Alert Notification is Absent</Ind>
         </Fun>
       </Expr>
       <Expr>
         <Fun
           uri=“dc:author”>
           <Ind>Randy Arvay</Ind>
         </Fun>
       </Expr>
       <Expr>
         <Fun
           uri=“dc:date”>
           <Ind>2/26/2009</Ind>
         </Fun>
       </Expr>
     </Plex>
   </label>
   <scope>
     <Ind
       uri=“#Information_Declassifier” />
   </scope>
   <oid>rule8</oid>
   <!--Alert Notification is Absent-->
```

```
      <if>
        <And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Present_Response</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“bool”>false</Ind>
            </rhs>
          </Equal>
          <Atom>
            <Rel>Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Query_Requires_Data_Insertion</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 19. RuleML code for RuleID 8

RuleID 9 is titled *Alert SD Secret IB*. This rule is used to verify that when an alert is present at the Secret level IB, intended for San Diego, and the requestor is at the Unclassified security level, that the *Alert_Present_Response* is set to *TRUE*. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert SD Secret IB</Ind>
            </Fun>
          </Expr>
          <Expr>
```

```
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule9</oid>
      <!--Alert SD Secret IB-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (SECRET IB) for Port of
San Diego”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Harbormaster San Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 20. RuleML code for RuleID 9

RuleID 10 is titled *EWO Query is Valid*. This is a filtering rule and is used to verify that the user fulfilling the EWO role is requesting an allowable query. It is checked against the actor’s allowable query set, to determine validity.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>EWO query is valid</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>3/5/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule10</oid>
  <!--EWO query is valid-->
  <if>
    <And>
      <And>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Role</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“EWO”</Ind>
          </rhs>
        </Equal>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Security_Level</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“Top Secret”</Ind>
          </rhs>
        </Equal>
      </And>
      <Atom>
        <Rel>PDP_Decision</Rel>
        <Var>Subject</Var>
      </Atom>
    </And>
  </if>
  <do>
    <Atom>
      <Rel>EWO_Valid_Query</Rel>
      <Var>Subject</Var>
      <Ind
        type=“bool”>true</Ind>
    </Atom>
    <Atom>
      <Rel>Vessel_Classification_Level</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
  </do>
</Rule>
```

Listing 21. RuleML code for RuleID 10

RuleID 11 is titled *IB Results 2*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Secret IB and Unclassified IB with ID Injection for Oakland Alert.” This is done after checking the user’s security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from both the Unclassified and Secret level IBs (since the user is at the Secret level), and then the required Alert Message insertion from the TS IB. This rule was created to allow for Harbormaster queries from the port of Oakland as an extension to the San Diego port accounted for in our scenario.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule11</oid>
      <!--IB Results 2-->
      <if>
        <And>
          <And>
```

```
            <And>
              <And>
                <Atom>
                  <Rel>Valid_Query</Rel>
                  <Var>Subject</Var>
                </Atom>
                <Atom>
                  <Rel>IB_Response_Wait</Rel>
                  <Var>Subject</Var>
                </Atom>
              </And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Secret"</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Alert Exists (TS IB) for Port of
Oakland"</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type="string">"Oakland"</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Secret IB and
Unclassified IB with ID Injection for Oakland Alert”</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 22. RuleML code for RuleID 11

RuleID 12 is titled *Negative ID Response to IB for Alert*. This rule provides an acknowledgment of a negative result for alert message presence. This is used to trigger other rules in the ruleset and to ensure that the ID returns a result in all cases. Without this rule, it is possible for the ID to return nothing, which allows for ambiguity in the ruleset. It checks if the query requires data insertion, which is sourced from the higher level IBs. If *FALSE*, then the IB that received the query is informed that its response to the user does not need to wait for the ID injection messaging prior to returning results to the user.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Negative ID Response to IB for Alert</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>2/26/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule12</oid>
      <!--Negative ID Response to IB for Alert-->
      <if>
        <And>
          <Atom>
            <Rel>Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Query_Requires_Data_Insertion</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“bool”>false</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>IB_Response_Wait</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 23. RuleML code for RuleID 12

RuleID 13 is titled *Oakland Harbormaster query is valid destination 2*. This rule is almost a duplicate of RuleID 4, but is used to support an alternative method of designating roles. In this case, if the role is listed as Harbormaster Oakland, instead of the generic Harbormaster designation, then the rule will process similarly. This is one of the rules used as extension to our primary scenario, which would allow for the port of Oakland queries to also work in the system. It is used to verify the user role, the role location, the PDP decision, and the type of query before deciding that it is a valid query and should be processed by the IB.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Oakland Harbormaster query is valid destination
2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule13</oid>
      <!--Oakland Harbormaster query is valid destination 2-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Harbormaster Oakland”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>HM_Queries</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Destination_Port Query”</Ind>
              </rhs>
            </Equal>
          </And>
          <Atom>
            <Rel>PDP_Decision</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
        <Atom>
          <Rel>User_Role_Location</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Oakland”</Ind>
        </Atom>
        <Atom>
          <Rel>Vessel_Classification_Level</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Security_Level</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
        <Atom>
          <Rel>Vessel_Destination_Port</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Role_Location</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
      </do>
    </Rule>
```

Listing 24. RuleML code for RuleID 13

RuleID 14 is titled *PDP Decision 2*. This is an externally sourced rule that would be completed by the Policy Decision Point service in our SOA system. It is used, based on the variables it is sourced with, to provide a decision for that user’s access to resources in the system. In this implementation, it is not referenced from an actual external PDP service, and this rule is intended to provide a positive response to the service check for a negative response.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Policy Decision Point 2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>2/26/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule14</oid>
      <!--Policy Decision Point 2-->
      <if>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Role_Permissions</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“bool”>false</Ind>
          </rhs>
        </Equal>
      </if>
      <do>
        <Atom>
          <Rel>PDP_Decision</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 25. RuleML code for RuleID 14

RuleID 15 is titled *Query is Invalid*. This is a filtering rule and is used to verify that the user (EWO or Harbormaster) is requesting an allowable query. It is sourced from two other rules, which check if the query against each actor’s allowable query set, to determine validity. This rule is intended to provide a positive response to the query check in the case of a negative response.

```
<Rule
   style=“active”
   evaluation=“strong”>
   <label>
     <Plex>
       <Expr>
         <Fun
           uri=“dc:title”>
           <Ind>Query Invalid</Ind>
         </Fun>
       </Expr>
       <Expr>
         <Fun
           uri=“dc:author”>
           <Ind>Randy Arvay</Ind>
         </Fun>
       </Expr>
       <Expr>
         <Fun
           uri=“dc:date”>
           <Ind>3/10/2009</Ind>
         </Fun>
       </Expr>
     </Plex>
   </label>
   <scope>
     <Ind
       uri=“#Information_Declassifier” />
   </scope>
   <oid>rule15</oid>
   <!--Query Invalid-->
   <if>
     <Or>
       <Equal>
         <lhs>
           <Atom>
             <Rel>HM_Valid_Query</Rel>
             <Var>Subject</Var>
           </Atom>
         </lhs>
         <rhs>
           <Ind
```

```
            type=“bool”>true</Ind>
        </rhs>
      </Equal>
      <Equal>
        <lhs>
          <Atom>
            <Rel>EWO_Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
        </lhs>
        <rhs>
          <Ind
            type=“bool”>true</Ind>
        </rhs>
      </Equal>
    </Or>
  </if>
  <do>
    <Atom>
      <Rel>Valid_Query</Rel>
      <Var>Subject</Var>
      <Ind
        type=“bool”>false</Ind>
    </Atom>
  </do>
</Rule>
```

Listing 26. RuleML code for RuleID 15

RuleID 16 is titled *Alert SD Secret IB 2*. This rule is used to verify that when an alert is present at the Secret level IB, intended for San Diego, and the requestor is at the Unclassified security level, that the *Alert_Present_Response* is set to *TRUE*. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level. This rule is equivalent to RuleID 9 and is used to support an alternative method of designating roles. In this case, if the role is listed as the generic Harbormaster, instead of the Harbormaster San Diego designation, then the rule will process similarly.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>Alert SD Secret IB 2</Ind>
```

```
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri="dc:author">
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri="dc:date">
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri="#Information_Declassifier" />
      </scope>
      <oid>rule16</oid>
      <!--Alert SD Secret IB 2-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Alert Exists (SECRET IB) for Port of
San Diego"</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Unclassified"</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
```

```
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Harbormaster"</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type="string">"San Diego"</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type="bool">true</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 27. RuleML code for RuleID 16

RuleID 17 is titled *IB Results 3*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of "Vessel Results from Secret IB and Unclassified IB." This is done after checking the user's security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from both the Unclassified and Secret level IB's (since the user is at the Secret level).

```
    <Rule
      style="active"
      evaluation="strong">
      <label>
```

```
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 3</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>2/26/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule17</oid>
      <!--IB Results 3-->
      <if>
        <And>
          <And>
            <Atom>
              <Rel>Valid_Query</Rel>
              <Var>Subject</Var>
            </Atom>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>IB_Response_Wait</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“bool”>false</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Security_Level</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
```

```
              <Ind
                type=“string”>“Secret”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Secret IB and
Unclassified IB”</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 28. RuleML code for RuleID 17

RuleID 18 is titled *Oakland Harbormaster query is valid origination*. This rule is supports an alternative method of designating roles. In this case, if the role is listed as Harbormaster Oakland, instead of the generic Harbormaster designation, then the rule will process similarly. This is one of the rules used as extension to our primary scenario, which would allow for the port of Oakland queries to also work in the system. It is used to verify the user role, the role location, the PDP decision, and that the query type is for an Originating Port before deciding that it is a valid query and should be processed by the IB.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Oakland Harbormaster query is valid
origination</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
      <Fun
        uri=“dc:date”>
        <Ind>3/5/2009</Ind>
      </Fun>
    </Expr>
  </Plex>
</label>
<scope>
  <Ind
    uri=“#Information_Declassifier” />
</scope>
<oid>rule18</oid>
<!--Oakland Harbormaster query is valid origination-->
<if>
  <And>
    <And>
      <Equal>
        <lhs>
          <Atom>
            <Rel>User_Role</Rel>
            <Var>Subject</Var>
          </Atom>
        </lhs>
        <rhs>
          <Ind
            type=“string”>“Harbormaster Oakland”</Ind>
        </rhs>
      </Equal>
      <Equal>
        <lhs>
          <Atom>
            <Rel>HM_Queries</Rel>
            <Var>Subject</Var>
          </Atom>
        </lhs>
        <rhs>
          <Ind
            type=“string”>“Originating_Port Query”</Ind>
        </rhs>
      </Equal>
    </And>
    <Atom>
      <Rel>PDP_Decision</Rel>
      <Var>Subject</Var>
    </Atom>
  </And>
</if>
<do>
  <Atom>
    <Rel>HM_Valid_Query</Rel>
    <Var>Subject</Var>
    <Ind
      type=“bool”>true</Ind>
  </Atom>
  <Atom>
```

```
      <Rel>User_Role_Location</Rel>
      <Var>Subject</Var>
      <Ind
        type="string">"Oakland"</Ind>
    </Atom>
    <Atom>
      <Rel>Vessel_Classification_Level</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
    <Atom>
      <Rel>Vessel_Originating_Port</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Role_Location</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
  </do>
</Rule>
```

Listing 29. RuleML code for RuleID 18

RuleID 19 is titled *Alert SD TS IB*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert Message against the security level of the user producing the query. This rule is used to verify that when an alert is present at the Top Secret level IB, intended for San Diego, the requestor is at the Unclassified or Secret security level, and that the *Alert_Present_Response* is set to *TRUE*. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert's classification level.

```
<Rule
  style="active"
  evaluation="strong">
  <label>
    <Plex>
      <Expr>
        <Fun
          uri="dc:title">
          <Ind>Alert SD TS IB</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri="dc:author">
```

```
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule19</oid>
      <!--Alert SD TS IB-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (TS IB) for Port of San
Diego”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Top Secret”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type="string">"Harbormaster San Diego"</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type="bool">true</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 30. RuleML code for RuleID 19

RuleID 20 is titled *EWO Query is Valid 2*. This is a filtering rule and is used to verify that the user fulfilling the EWO role is requesting an allowable query. It is checked against the actor's allowable query set, to determine validity. This rule is equivalent to RuleID 10, but accounts for a different role naming convention. In this case, the user's role is Electronic Warfare Officer, instead of EWO.

```
<Rule
  style="active"
  evaluation="strong">
  <label>
    <Plex>
      <Expr>
        <Fun
          uri="dc:title">
          <Ind>EWO query is valid 2</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri="dc:author">
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri="dc:date">
          <Ind>3/5/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
```

```
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule20</oid>
  <!--EWO query is valid 2-->
  <if>
    <And>
      <And>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Role</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“Electronic Warfare Officer”</Ind>
          </rhs>
        </Equal>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Security_Level</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“Top Secret”</Ind>
          </rhs>
        </Equal>
      </And>
      <Atom>
        <Rel>PDP_Decision</Rel>
        <Var>Subject</Var>
      </Atom>
    </And>
  </if>
  <do>
    <Atom>
      <Rel>EWO_Valid_Query</Rel>
      <Var>Subject</Var>
      <Ind
        type=“bool”>true</Ind>
    </Atom>
    <Atom>
      <Rel>Vessel_Classification_Level</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
  </do>
</Rule>
```

Listing 31. RuleML code for RuleID 20

RuleID 21 is titled *IB Results 4*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Unclassified IB.” This is done after checking the user’s security level and for the presence of an alert for this location being *FALSE*. The IB, in this case, would return its regular result from the Unclassified-level IB.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>IB Results 4</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>2/26/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule21</oid>
  <!--IB Results 4-->
  <if>
    <And>
      <And>
        <Atom>
          <Rel>Valid_Query</Rel>
          <Var>Subject</Var>
        </Atom>
        <Equal>
          <lhs>
            <Atom>
              <Rel>IB_Response_Wait</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“bool”>false</Ind>
          </rhs>
        </Equal>
      </And>
      <Equal>
        <lhs>
          <Atom>
            <Rel>User_Security_Level</Rel>
            <Var>Subject</Var>
          </Atom>
        </lhs>
        <rhs>
          <Ind
            type=“string”>“Unclassified”</Ind>
        </rhs>
      </Equal>
    </And>
  </if>
  <do>
    <Atom>
      <Rel>Return_IB_Results</Rel>
      <Var>Subject</Var>
      <Ind
        type=“string”>“Vessel Results from Unclassified IB”</Ind>
    </Atom>
  </do>
</Rule>
```

Listing 32. RuleML code for RuleID 21

RuleID 22 is titled *Oakland Harbormaster query is valid origination 2*. This rule is almost a duplicate of RuleID 18, but is used to support an alternative method of designating roles. In this case, if the role is listed as the generic Harbormaster, instead of the Harbormaster Oakland designation, then the rule will process similarly. This is one of the rules used as extension to our primary scenario, which would allow for the port of Oakland queries to also work in the system. It is used to verify the user role, the role location, the PDP decision, and the type of query is an originating port query before deciding that it is a valid query and should be processed by the IB.

```
    <Rule
      style="active"
      evaluation="strong">
      <label>
        <Plex>
          <Expr>
            <Fun
              uri="dc:title">
              <Ind>Oakland Harbormaster query is valid origination
2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri="dc:author">
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri="dc:date">
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri="#Information_Declassifier" />
      </scope>
      <oid>rule22</oid>
      <!--Oakland Harbormaster query is valid origination 2-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Harbormaster"</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role_Location</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Oakland”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>HM_Queries</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Originating_Port Query”</Ind>
              </rhs>
            </Equal>
          </And>
          <Atom>
            <Rel>PDP_Decision</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
        <Atom>
          <Rel>Vessel_Classification_Level</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Security_Level</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
        <Atom>
          <Rel>Vessel_Originating_Port</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Role_Location</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
      </do>
    </Rule>
```

Listing 33. RuleML code for RuleID 22

RuleID 23 is titled *Alert SD TS IB*. This rule is used to verify that when an alert is present at the Top Secret level IB, intended for San Diego, and the requestor is at the Unclassified or Secret security level, that the *Alert_Present_Response* is set to *TRUE*. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level. This rule is equivalent to RuleID 19 and is used to support an alternative method of designating roles. In this case, if the role is listed as the generic Harbormaster, instead of the Harbormaster San Diego designation, then the rule will process similarly.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert SD TS IB 2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule23</oid>
      <!--Alert SD TS IB 2-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Alert Exists (TS IB) for Port of San
Diego"</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Top Secret"</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Harbormaster"</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type="string">"San Diego"</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type="bool">true</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 34. RuleML code for RuleID 23

RuleID 24 is titled *IB Results 5*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from TS IB, Secret IB and Unclassified IB.” This is done after checking the user’s security level is equal to *Top Secret*. The user in this case is authorized all data from each of the classification levels and does not require injection to the IB results.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 5</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule24</oid>
      <!--IB Results 5-->
      <if>
        <And>
          <And>
```

```
            <Atom>
              <Rel>Valid_Query</Rel>
              <Var>Subject</Var>
            </Atom>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>IB_Response_Wait</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“bool”>false</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Security_Level</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Top Secret”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from TS IB, Secret IB, and
Unclassified IB”</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 35. RuleML code for RuleID 24

RuleID 25 is titled *SD Harbormaster query is valid destination*. This rule is used to verify the user role as *Harbormaster*, the role location is *San Diego*, the PDP decision is *TRUE*, and the query is both valid and of the type *Destination_Port_Query* before deciding that it is valid and should be processed by the IB.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>SD Harbormaster query is valid destination</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule25</oid>
      <!--SD Harbormaster query is valid destination-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Harbormaster”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role_Location</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“San Diego”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>HM_Queries</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Destination_Port Query"</Ind>
              </rhs>
            </Equal>
          </And>
          <Atom>
            <Rel>PDP_Decision</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type="bool">true</Ind>
        </Atom>
        <Atom>
          <Rel>Vessel_Classification_Level</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Security_Level</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
        <Atom>
          <Rel>Vessel_Destination_Port</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Role_Location</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
      </do>
    </Rule>
```

Listing 36. RuleML code for RuleID 25

RuleID 26 is titled *Alert Oak Secret IB*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert

Message against the security level of the user producing the query. This particular rule is used as an extension to our baseline scenario to account for users at the port of Oakland. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert Oak Secret IB</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule26</oid>
      <!--Alert Oak Secret IB-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (SECRET IB) for Port of
Oakland”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Harbormaster Oakland”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 37. RuleML code for RuleID 26

RuleID 27 is titled *IB Results 6*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rulebase with a sample query. In this case, it returns the IB result of “Invalid Query!.” This is done after checking the query validity from the *Valid_Query* variable.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
```

```
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 6</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule27</oid>
      <!--IB Results 6-->
      <if>
        <Equal>
          <lhs>
            <Atom>
              <Rel>Valid_Query</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“bool”>false</Ind>
          </rhs>
        </Equal>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Invalid Query!”</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 38. RuleML code for RuleID 27

RuleID 28 is titled *SD Harbormaster query is valid destination 2*. This rule is identical in functionality to RuleID 25, but supports the extended role convention using *Harbormaster San Diego* instead of the generic *Harbormaster* role. This rule is used to verify the user role as *Harbormaster San Diego*, the PDP decision is *TRUE*, and the type of query is *Destination_Port_Query* before deciding that it is valid and should be processed by the IB.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>SD Harbormaster query is valid destination 2</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>3/5/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule28</oid>
  <!--SD Harbormaster query is valid destination 2-->
  <if>
    <And>
      <And>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Role</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
```

```
                type=“string”>“Harbormaster San Diego”</Ind>
            </rhs>
          </Equal>
          <Equal>
            <lhs>
              <Atom>
                <Rel>HM_Queries</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Destination_Port Query”</Ind>
            </rhs>
          </Equal>
        </And>
        <Atom>
          <Rel>PDP_Decision</Rel>
          <Var>Subject</Var>
        </Atom>
      </And>
    </if>
    <do>
      <Atom>
        <Rel>HM_Valid_Query</Rel>
        <Var>Subject</Var>
        <Ind
          type=“bool”>true</Ind>
      </Atom>
      <Atom>
        <Rel>User_Role_Location</Rel>
        <Var>Subject</Var>
        <Ind
          type=“string”>“San Diego”</Ind>
      </Atom>
      <Atom>
        <Rel>Vessel_Classification_Level</Rel>
        <Var>Subject</Var>
        <Atom>
          <Rel>User_Security_Level</Rel>
          <Var>Subject</Var>
        </Atom>
      </Atom>
      <Atom>
        <Rel>Vessel_Destination_Port</Rel>
        <Var>Subject</Var>
        <Atom>
          <Rel>User_Role_Location</Rel>
          <Var>Subject</Var>
        </Atom>
      </Atom>
    </do>
  </Rule>
```

Listing 39. RuleML code for RuleID 28

RuleID 29 is titled *Alert Oak Secret IB 2*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert Message against the security level of the user producing the query. It is identical in functionality to RuleID 26, but supports the generic role moniker of *Harbormaster* instead of *Harbormaster Oakland*. This particular rule is used as an extension to our baseline scenario to account for users at the port of Oakland. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert Oak Secret IB 2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule29</oid>
      <!--Alert Oak Secret IB 2-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
```

```
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Alert Exists (SECRET IB) for Port of
Oakland"</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Unclassified"</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Harbormaster"</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type="string">"Oakland"</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
      type=“bool”>true</Ind>
    </Atom>
   </do>
  </Rule>
```

Listing 40. RuleML code for RuleID 29

RuleID 30 is titled *SD Harbormaster query is valid origination*. This rule is identical in functionality to RuleID 32, but supports the extended role convention using *Harbormaster San Diego* instead of the generic *Harbormaster* role. This rule is used to verify the user role as *Harbormaster San Diego*, the PDP decision is *TRUE*, and the type of query is *Originating_Port_Query* before deciding that it is valid and should be processed by the IB. This rule checks the query that is being processed and sets conditions to restrict the IB response as a result. This rule is designed to eliminate the user’s ability to repeatedly query the system to troll for information, or the lack of information, again a key element in preventing the Misuse Case. In the original system, queries were directly processed by the IB after entry. In the re-architected system, we reroute all queries through the ID (per Figure 26) where we begin the new data flow at continuation point I instead of direct processing by the IB in the original flow. In this case, the query and resultant data must include the Harbormaster’s port (either Origination or Destination) for details to be returned by the IB and ID. This is part of the <Detect> in the Security Use Case that we had to reroute queries in the system. This will prevent a Harbormaster from continually querying the system to get information about vessels from the system that do not directly pertain to his or her role (at his location) and his normal performance of duty. This rule verifies that the user is a Harbormaster from San Diego and conducting an Originating Port Query, which he is authorized because of his role’s duties and obligations, as well as a PDP service check to validate the role’s permissions to execute a basic query and to access information from the IB. The rule then establishes this as a valid query for the San Diego Harbormaster, while it also restricts the query response by limiting Vessel_Classification_Level to the User_Security_Level, and setting the field for Vessel_Originating_Port to the User_Role_Location, or in this case San Diego.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>SD Harbormaster query is valid origination</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>3/5/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule30</oid>
  <!--SD Harbormaster query is valid origination-->
  <if>
    <And>
      <And>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Role</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“Harbormaster San Diego”</Ind>
          </rhs>
        </Equal>
        <Equal>
          <lhs>
            <Atom>
              <Rel>HM_Queries</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
```

```
         type="string">"Originating_Port Query"</Ind>
       </rhs>
      </Equal>
     </And>
     <Atom>
      <Rel>PDP_Decision</Rel>
      <Var>Subject</Var>
     </Atom>
    </And>
   </if>
   <do>
    <Atom>
     <Rel>HM_Valid_Query</Rel>
     <Var>Subject</Var>
     <Ind
      type="bool">true</Ind>
    </Atom>
    <Atom>
     <Rel>User_Role_Location</Rel>
     <Var>Subject</Var>
     <Ind
      type="string">"San Diego"</Ind>
    </Atom>
    <Atom>
     <Rel>Vessel_Classification_Level</Rel>
     <Var>Subject</Var>
     <Atom>
      <Rel>User_Security_Level</Rel>
      <Var>Subject</Var>
     </Atom>
    </Atom>
    <Atom>
     <Rel>Vessel_Originating_Port</Rel>
     <Var>Subject</Var>
     <Atom>
      <Rel>User_Role_Location</Rel>
      <Var>Subject</Var>
     </Atom>
    </Atom>
   </do>
  </Rule>
```

Listing 41. RuleML code for RuleID 30

RuleID 31 is titled *Alert Oak TS IB*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert Message against the security level of the user producing the query. It is identical in functionality to RuleID 33, but supports the generic role moniker of *Harbormaster* instead of *Harbormaster Oakland*. This particular rule is used as an extension to our baseline scenario to account for users at the port of Oakland. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert's classification level.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert OakTS IB</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule31</oid>
      <!--Alert OakTS IB-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Alert Exists (TS IB) for Port of
Oakland”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Top Secret”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Harbormaster”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Oakland”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 42. RuleML code for RuleID 31

RuleID 32 is titled *SD Harbormaster query is valid origination 2*. This rule is almost a duplicate of RuleID 30, but is used to support an alternative method of designating roles. In this case, if the role is listed as the generic *Harbormaster*, instead of the *Harbormaster*

San Diego designation, then the rule will process similarly. This rule checks the query that is being processed and sets conditions to restrict the IB response as a result. This rule is designed to eliminate the user's ability to repeatedly query the system to troll for information, or the lack of information, again a key element in preventing the Misuse Case. In the original system, queries were directly processed by the IB after entry. In the re-architected system, we reroute all queries through the ID (per Figure 26) where we begin the new data flow at continuation point I instead of direct processing by the IB in the original flow. In this case, the query and resultant data must include the Harbormaster's port (either Origination or Destination) for details to be returned by the IB and ID. This is part of the <Detect> in the Security Use Case that we had to reroute queries in the system. This will prevent a Harbormaster from continually querying the system to get information about vessels from the system that do not directly pertain to his or her role (at his location) and his normal performance of duty. This rule verifies that the user is a Harbormaster from San Diego and conducting an Originating Port Query, which he is authorized because of his role's duties and obligations, as well as a PDP service check to validate the role's permissions to execute a basic query and to access information from the IB. The rule then establishes this as a valid query for the San Diego Harbormaster, while it also restricts the query response by limiting Vessel_Classification_Level to the User_Security_Level, and setting the field for Vessel_Originating_Port to the User_Role_Location, or in this case San Diego.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>SD Harbormaster query is valid origination 2</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
```

```
   <Fun
    uri=“dc:date”>
    <Ind>3/5/2009</Ind>
   </Fun>
  </Expr>
 </Plex>
</label>
<scope>
 <Ind
  uri=“#Information_Declassifier” />
</scope>
<oid>rule32</oid>
<!--SD Harbormaster query is valid origination 2-->
<if>
 <And>
  <And>
   <And>
    <Equal>
     <lhs>
      <Atom>
       <Rel>User_Role</Rel>
       <Var>Subject</Var>
      </Atom>
     </lhs>
     <rhs>
      <Ind
       type=“string”>“Harbormaster”</Ind>
     </rhs>
    </Equal>
    <Equal>
     <lhs>
      <Atom>
       <Rel>User_Role_Location</Rel>
       <Var>Subject</Var>
      </Atom>
     </lhs>
     <rhs>
      <Ind
       type=“string”>“San Diego”</Ind>
     </rhs>
    </Equal>
   </And>
   <Equal>
    <lhs>
     <Atom>
      <Rel>HM_Queries</Rel>
      <Var>Subject</Var>
     </Atom>
    </lhs>
    <rhs>
     <Ind
      type=“string”>“Originating_Port Query”</Ind>
    </rhs>
   </Equal>
  </And>
```

```
          <Atom>
            <Rel>PDP_Decision</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
        <Atom>
          <Rel>Vessel_Classification_Level</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Security_Level</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
        <Atom>
          <Rel>Vessel_Originating_Port</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Role_Location</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
      </do>
    </Rule>
```

Listing 43. RuleML code for RuleID 32

RuleID 33 is titled *Alert Oak TS IB 2*. This rule is established as a trigger within the ruleset to identify that an Alert Message is present and to verify the level of the Alert Message against the security level of the user producing the query. This rule is identical in functionality to RuleID 30. This particular rule is used as an extension to our baseline scenario to account for users at the port of Oakland. This will indicate to the ID and IB that an alert is present that pertains to the user query, and that the user is below the alert’s classification level.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
      <Fun
       uri=“dc:title”>
       <Ind>Alert OakTS IB 2</Ind>
      </Fun>
     </Expr>
     <Expr>
      <Fun
       uri=“dc:author”>
       <Ind>Randy Arvay</Ind>
      </Fun>
     </Expr>
     <Expr>
      <Fun
       uri=“dc:date”>
       <Ind>3/5/2009</Ind>
      </Fun>
     </Expr>
    </Plex>
   </label>
   <scope>
    <Ind
     uri=“#Information_Declassifier” />
   </scope>
   <oid>rule33</oid>
   <!--Alert OakTS IB 2-->
   <if>
    <And>
     <And>
      <Equal>
       <lhs>
        <Atom>
         <Rel>Alert_Notification</Rel>
         <Var>Subject</Var>
        </Atom>
       </lhs>
       <rhs>
        <Ind
         type=“string”>“Alert Exists (TS IB) for Port of
Oakland”</Ind>
       </rhs>
      </Equal>
      <Equal>
       <lhs>
        <Atom>
         <Rel>User_Security_Level</Rel>
         <Var>Subject</Var>
        </Atom>
       </lhs>
       <rhs>
        <Ind
         type=“string”>“Top Secret”</Ind>
       </rhs>
      </Equal>
     </And>
     <Equal>
```

```
        <lhs>
          <Atom>
            <Rel>User_Role</Rel>
            <Var>Subject</Var>
          </Atom>
        </lhs>
        <rhs>
          <Ind
            type=“string”>“Harbormaster Oakland”</Ind>
        </rhs>
      </Equal>
    </And>
  </if>
  <do>
    <Atom>
      <Rel>Alert_Present_Response</Rel>
      <Var>Subject</Var>
      <Ind
        type=“bool”>true</Ind>
    </Atom>
  </do>
</Rule>
```

Listing 44. RuleML code for RuleID 33

RuleID 34 is titled *IB Results 0.1*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Unclassified IB with ID Injection for Oakland Alert.” This is done after checking the user’s security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from the Unclassified level IB (since the user is at the Unclassified level), and then the required Alert Message insertion from the higher (TS or Secret) IB. The level of the Alert Message is not revealed to the user from the IB. This rule was created to allow for Harbormaster queries from the port of Oakland as an extension to the San Diego port accounted for in our scenario.

```
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>IB Results 0.1</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>3/5/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule34</oid>
  <!--IB Results 0.1-->
  <if>
    <And>
      <And>
        <And>
          <Atom>
            <Rel>Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
          <Atom>
            <Rel>IB_Response_Wait</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Security_Level</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“Unclassified”</Ind>
          </rhs>
        </Equal>
      </And>
      <Equal>
        <lhs>
          <Atom>
            <Rel>Alert_Notification</Rel>
            <Var>Subject</Var>
          </Atom>
        </lhs>
        <rhs>
          <Ind
            type=“string”>“Alert Exists (SECRET IB) for Port of
  Oakland”</Ind>
        </rhs>
       </Equal>
      </And>
     </if>
     <do>
      <Atom>
       <Rel>Return_IB_Results</Rel>
       <Var>Subject</Var>
       <Ind
        type=“string”>“Vessel Results from Unclassified IB with ID
  Injection for Oakland Alert”</Ind>
      </Atom>
     </do>
    </Rule>
```

Listing 45. RuleML code for RuleID 34

RuleID 35 is titled *IB Results 0.2*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Unclassified IB with ID Injection for San Diego Alert.” This is done after checking the user’s security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from the Unclassified level IB (since the user is at the Unclassified level), and then the required Alert Message insertion from the higher (TS or Secret) IB. The level of the Alert Message is not revealed to the user from the IB.

```
  <Rule
   style=“active”
   evaluation=“strong”>
   <label>
    <Plex>
     <Expr>
      <Fun
       uri=“dc:title”>
       <Ind>IB Results 0.2</Ind>
      </Fun>
     </Expr>
     <Expr>
      <Fun
       uri=“dc:author”>
```

```
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri="dc:date">
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri="#Information_Declassifier" />
      </scope>
      <oid>rule35</oid>
      <!--IB Results 0.2-->
      <if>
        <And>
          <And>
            <And>
              <Atom>
                <Rel>Valid_Query</Rel>
                <Var>Subject</Var>
              </Atom>
              <Atom>
                <Rel>IB_Response_Wait</Rel>
                <Var>Subject</Var>
              </Atom>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Unclassified"</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Notification</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type="string">"Alert Exists (SECRET IB) for Port of San
Diego"</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Unclassified IB with ID
Injection for San Diego Alert”</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 46. RuleML code for RuleID 35

RuleID 36 is titled *IB Results 0.3*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Unclassified IB with ID Injection for San Diego Alert.” This is done after checking the user’s security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from the Unclassified level IB (since the user is at the Unclassified level), and then the required Alert Message insertion from the TS IB. The level of the Alert Message is not revealed to the user from the IB.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 0.3</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule36</oid>
      <!--IB Results 0.3-->
      <if>
        <And>
          <And>
            <And>
              <Atom>
                <Rel>Valid_Query</Rel>
                <Var>Subject</Var>
              </Atom>
              <Atom>
                <Rel>IB_Response_Wait</Rel>
                <Var>Subject</Var>
              </Atom>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Notification</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Alert Exists (TS IB) for Port of San
Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Unclassified IB with ID
Injection for San Diego Alert”</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 47. RuleML code for RuleID 36

RuleID 37 is titled *No Alert IB*. This rule is established as a trigger within the ruleset to identify that an Alert Message is not present at a higher level IB. This is provide a positive response to a negative search result (i.e., no alert message found).

```
   <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>No Alert IB</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/10/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule37</oid>
      <!--No Alert IB-->
      <if>
        <Equal>
          <lhs>
            <Atom>
              <Rel>Alert_Notification</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“No Alert Exists”</Ind>
          </rhs>
        </Equal>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 48. RuleML code for RuleID 37

RuleID 38 is titled *IB Results 2.1*. This is one of the ten IB results rules that is intended to replicate the results from an actual IB. It is used, based on the variables it is sourced with to provide an expected result when testing the rule base with a sample query. In this case, it returns the IB result of “Vessel Results from Secret IB and Unclassified IB with ID Injection for San Diego Alert.” This is done after checking the user’s security level and for the presence of an alert for this location. The IB, in this case, would return its regular result from both the Unclassified and Secret level IB’s (since the user is at the Secret level), and then the required Alert Message insertion from the TS IB.

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 2.1</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule38</oid>
      <!--IB Results 2.1-->
      <if>
        <And>
          <And>
            <And>
              <And>
                <Atom>
                  <Rel>Valid_Query</Rel>
                  <Var>Subject</Var>
                </Atom>
                <Atom>
                  <Rel>IB_Response_Wait</Rel>
                  <Var>Subject</Var>
                </Atom>
              </And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Secret”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (TS IB) for Port of San
Diego”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
```

```
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“San Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Secret IB and
Unclassified IB with ID Injection for San Diego Alert”</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 49. RuleML code for RuleID 38

RuleID 39 is titled *EWO Query is Invalid.* This is a filtering rule and is used to verify that the user (EWO) is requesting an allowable query. It verifies the role of the query requestor and is intended to provide a positive response to the query check in the case of a negative response (i.e., an invalid query request).

```
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>EWO query is invalid</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/10/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule39</oid>
      <!--EWO query is invalid-->
      <if>
        <Or>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Electronic Warfare Officer”</Ind>
            </rhs>
          </Equal>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“EWO”</Ind>
            </rhs>
          </Equal>
        </Or>
      </if>
      <do>
        <Atom>
          <Rel>EWO_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
```

Listing 50. RuleML code for RuleID 39

RuleID 40 is titled *HM Query is Invalid.* This is a filtering rule and is used to verify that the user (Harbormaster) is requesting an allowable query. In this case, it allows for Harbormaster queries to originate from San Diego and Oakland only. Any other location for the role of a Harbormaster will make the query invalid. It verifies the role of the query requestor and is intended to provide a positive response to the query check in the case of a negative response (i.e., an invalid query request).

```
<Rule
   style=“active”
   evaluation=“strong”>
   <label>
     <Plex>
       <Expr>
         <Fun
           uri=“dc:title”>
           <Ind>HM query is invalid</Ind>
         </Fun>
       </Expr>
       <Expr>
         <Fun
           uri=“dc:author”>
           <Ind>Randy Arvay</Ind>
         </Fun>
       </Expr>
       <Expr>
         <Fun
           uri=“dc:date”>
           <Ind>3/10/2009</Ind>
         </Fun>
       </Expr>
     </Plex>
   </label>
   <scope>
     <Ind
       uri=“#Information_Declassifier” />
   </scope>
   <oid>rule40</oid>
   <!--HM query is invalid-->
   <if>
     <Or>
       <Equal>
         <lhs>
           <Atom>
             <Rel>User_Role</Rel>
             <Var>Subject</Var>
           </Atom>
         </lhs>
         <rhs>
           <Ind
             type=“string”>“Harbormaster Oakland”</Ind>
         </rhs>
       </Equal>
       <Equal>
         <lhs>
           <Atom>
             <Rel>User_Role</Rel>
```

```
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Harbormaster San Diego”</Ind>
            </rhs>
          </Equal>
        </Or>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
  </Rulebase>
</RuleML>
```

Listing 51. RuleML code for RuleID 40

APPENDIX B. RULEML CODE

```
<?xml version=“1.0” encoding=“utf-8”?>
<RuleML xmlns=“http://www.ruleml.org/0.91/xsd”>
  <Rulebase>
    <label>
      <Plex>
        <Expr>
          <Fun
            uri=“dc:title”>
            <Ind>MDA_Scenario_Arvay_current</Ind>
          </Fun>
        </Expr>
        <Expr>
          <Fun
            uri=“dc:author”>
            <Ind>Randy Arvay</Ind>
          </Fun>
        </Expr>
        <Expr>
          <Fun
            uri=“dc:date”>
            <Ind>12/4/2008</Ind>
          </Fun>
        </Expr>
      </Plex>
    </label>
    <!--Rule Policy Information_Declassifier-->
    <!--oid of the rule base / module -->
    <Ind>Information_Declassifier</Ind>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert Not Valid for Role Location</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/10/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule0</oid>
      <!--Alert Not Valid for Role Location-->
      <if>
        <Or>
          <Or>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (SECRET IB) for Port of
Los Angeles”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (TS IB) for Port of Los
Angeles”</Ind>
              </rhs>
            </Equal>
          </Or>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Notification</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“No Alert Exists”</Ind>
            </rhs>
          </Equal>
        </Or>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert Notification is Present</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>2/26/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule1</oid>
      <!--Alert Notification is Present-->
      <if>
        <Equal>
          <lhs>
            <Atom>
              <Rel>Alert_Present_Response</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“bool”>true</Ind>
          </rhs>
        </Equal>
      </if>
      <do>
        <Atom>
          <Rel>Query_Requires_Data_Insertion</Rel>
          <Var>Subject</Var>
```

```
            <Ind
              type=“bool”>true</Ind>
          </Atom>
        </do>
      </Rule>
      <Rule
        style=“active”
        evaluation=“strong”>
        <label>
          <Plex>
            <Expr>
              <Fun
                uri=“dc:title”>
                <Ind>Est Location for Role</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:author”>
                <Ind>Randy Arvay</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:date”>
                <Ind>3/5/2009</Ind>
              </Fun>
            </Expr>
          </Plex>
        </label>
        <scope>
          <Ind
            uri=“#Information_Declassifier” />
        </scope>
        <oid>rule2</oid>
        <!--Est Location for Role-->
        <if>
          <Or>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Location</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Oakland”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Location</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“San Diego”</Ind>
              </rhs>
            </Equal>
          </Or>
        </if>
        <do>
          <Atom>
            <Rel>User_Role_Location</Rel>
            <Var>Subject</Var>
            <Atom>
              <Rel>Location</Rel>
              <Var>Subject</Var>
            </Atom>
          </Atom>
        </do>
      </Rule>
      <Rule
        style=“active”
        evaluation=“strong”>
        <label>
          <Plex>
            <Expr>
              <Fun
                uri=“dc:title”>
                <Ind>IB Results 0</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:author”>
                <Ind>Randy Arvay</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:date”>
                <Ind>3/5/2009</Ind>
              </Fun>
            </Expr>
          </Plex>
        </label>
        <scope>
          <Ind
            uri=“#Information_Declassifier” />
        </scope>
        <oid>rule3</oid>
        <!--IB Results 0-->
        <if>
          <And>
            <And>
              <And>
```

```
              <Atom>
                <Rel>Valid_Query</Rel>
                <Var>Subject</Var>
              </Atom>
              <Atom>
                <Rel>IB_Response_Wait</Rel>
                <Var>Subject</Var>
              </Atom>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Notification</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Alert Exists (TS IB) for Port of
Oakland”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Unclassified IB with ID
Injection for Oakland Alert”</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Oakland Harbormaster query is valid
destination</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule4</oid>
      <!--Oakland Harbormaster query is valid destination-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Harbormaster”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role_Location</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Oakland”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
```

```
                <Atom>
                  <Rel>HM_Queries</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Destination_Port Query”</Ind>
              </rhs>
            </Equal>
          </And>
          <Atom>
            <Rel>PDP_Decision</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
        <Atom>
          <Rel>Vessel_Classification_Level</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Security_Level</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
        <Atom>
          <Rel>Vessel_Destination_Port</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Role_Location</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Policy Decision Point</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
```

```
                  uri=“dc:author”>
                  <Ind>Randy Arvay</Ind>
                </Fun>
              </Expr>
              <Expr>
                <Fun
                  uri=“dc:date”>
                  <Ind>2/26/2009</Ind>
                </Fun>
              </Expr>
            </Plex>
          </label>
          <scope>
            <Ind
              uri=“#Information_Declassifier” />
          </scope>
          <oid>rule5</oid>
          <!--Policy Decision Point-->
          <if>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role_Permissions</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“bool”>true</Ind>
              </rhs>
            </Equal>
          </if>
          <do>
            <Atom>
              <Rel>PDP_Decision</Rel>
              <Var>Subject</Var>
              <Ind
                type=“bool”>true</Ind>
            </Atom>
          </do>
        </Rule>
        <Rule
          style=“active”
          evaluation=“strong”>
          <label>
            <Plex>
              <Expr>
                <Fun
                  uri=“dc:title”>
                  <Ind>Positive ID Response to IB for Alert</Ind>
                </Fun>
              </Expr>
              <Expr>
                <Fun
                  uri=“dc:author”>
```

```
     <Ind>Randy Arvay</Ind>
    </Fun>
   </Expr>
   <Expr>
    <Fun
     uri=“dc:date”>
     <Ind>12/5/2008</Ind>
    </Fun>
   </Expr>
  </Plex>
 </label>
 <scope>
  <Ind
   uri=“#Information_Declassifier” />
 </scope>
 <oid>rule6</oid>
 <!--Positive ID Response to IB for Alert-->
 <if>
  <And>
   <Atom>
    <Rel>Valid_Query</Rel>
    <Var>Subject</Var>
   </Atom>
   <Atom>
    <Rel>Query_Requires_Data_Insertion</Rel>
    <Var>Subject</Var>
   </Atom>
  </And>
 </if>
 <do>
  <Atom>
   <Rel>IB_Response_Wait</Rel>
   <Var>Subject</Var>
   <Ind
    type=“bool”>true</Ind>
  </Atom>
 </do>
</Rule>
<Rule
 style=“active”
 evaluation=“strong”>
 <label>
  <Plex>
   <Expr>
    <Fun
     uri=“dc:title”>
     <Ind>Query is Valid</Ind>
    </Fun>
   </Expr>
   <Expr>
    <Fun
     uri=“dc:author”>
     <Ind>Randy Arvay</Ind>
    </Fun>
   </Expr>
```

```
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/10/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule7</oid>
      <!--Query is Valid-->
      <if>
        <Or>
          <Atom>
            <Rel>HM_Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
          <Atom>
            <Rel>EWO_Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
        </Or>
      </if>
      <do>
        <Atom>
          <Rel>Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert Notification is Absent</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>2/26/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule8</oid>
      <!--Alert Notification is Absent-->
      <if>
        <And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Present_Response</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“bool”>false</Ind>
            </rhs>
          </Equal>
          <Atom>
            <Rel>Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Query_Requires_Data_Insertion</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert SD Secret IB</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
```

```
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule9</oid>
      <!--Alert SD Secret IB-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (SECRET IB) for Port of
San Diego”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Harbormaster San Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>EWO query is valid</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule10</oid>
      <!--EWO query is valid-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
```

```
            <Ind
              type=“string”>“EWO”</Ind>
          </rhs>
        </Equal>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Security_Level</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“Top Secret”</Ind>
          </rhs>
        </Equal>
      </And>
      <Atom>
        <Rel>PDP_Decision</Rel>
        <Var>Subject</Var>
      </Atom>
    </And>
  </if>
  <do>
    <Atom>
      <Rel>EWO_Valid_Query</Rel>
      <Var>Subject</Var>
      <Ind
        type=“bool”>true</Ind>
    </Atom>
    <Atom>
      <Rel>Vessel_Classification_Level</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
  </do>
</Rule>
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>IB Results 2</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
```

```
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule11</oid>
      <!--IB Results 2-->
      <if>
        <And>
          <And>
            <And>
              <And>
                <Atom>
                  <Rel>Valid_Query</Rel>
                  <Var>Subject</Var>
                </Atom>
                <Atom>
                  <Rel>IB_Response_Wait</Rel>
                  <Var>Subject</Var>
                </Atom>
              </And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Secret”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (TS IB) for Port of
Oakland”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Oakland”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Secret IB and
Unclassified IB with ID Injection for Oakland Alert”</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Negative ID Response to IB for Alert</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>2/26/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule12</oid>
      <!--Negative ID Response to IB for Alert-->
      <if>
        <And>
          <Atom>
            <Rel>Valid_Query</Rel>
            <Var>Subject</Var>
          </Atom>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Query_Requires_Data_Insertion</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“bool”>false</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>IB_Response_Wait</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Oakland Harbormaster query is valid destination
2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
```

```
  </Plex>
</label>
<scope>
  <Ind
    uri=“#Information_Declassifier” />
</scope>
<oid>rule13</oid>
<!--Oakland Harbormaster query is valid destination 2-->
<if>
  <And>
    <And>
      <Equal>
        <lhs>
          <Atom>
            <Rel>User_Role</Rel>
            <Var>Subject</Var>
          </Atom>
        </lhs>
        <rhs>
          <Ind
            type=“string”>“Harbormaster Oakland”</Ind>
        </rhs>
      </Equal>
      <Equal>
        <lhs>
          <Atom>
            <Rel>HM_Queries</Rel>
            <Var>Subject</Var>
          </Atom>
        </lhs>
        <rhs>
          <Ind
            type=“string”>“Destination_Port Query”</Ind>
        </rhs>
      </Equal>
    </And>
    <Atom>
      <Rel>PDP_Decision</Rel>
      <Var>Subject</Var>
    </Atom>
  </And>
</if>
<do>
  <Atom>
    <Rel>HM_Valid_Query</Rel>
    <Var>Subject</Var>
    <Ind
      type=“bool”>true</Ind>
  </Atom>
  <Atom>
    <Rel>User_Role_Location</Rel>
    <Var>Subject</Var>
    <Ind
      type=“string”>“Oakland”</Ind>
  </Atom>
```

```
    <Atom>
      <Rel>Vessel_Classification_Level</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
    <Atom>
      <Rel>Vessel_Destination_Port</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Role_Location</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
  </do>
</Rule>
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>Policy Decision Point 2</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>2/26/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule14</oid>
  <!--Policy Decision Point 2-->
  <if>
    <Equal>
      <lhs>
        <Atom>
          <Rel>User_Role_Permissions</Rel>
          <Var>Subject</Var>
        </Atom>
      </lhs>
      <rhs>
        <Ind
          type=“bool”>false</Ind>
      </rhs>
    </Equal>
  </if>
  <do>
    <Atom>
      <Rel>PDP_Decision</Rel>
      <Var>Subject</Var>
      <Ind
        type=“bool”>false</Ind>
    </Atom>
  </do>
</Rule>
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>Query Invalid</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>3/10/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule15</oid>
  <!--Query Invalid-->
  <if>
    <Or>
      <Equal>
        <lhs>
          <Atom>
            <Rel>HM_Valid_Query</Rel>
            <Var>Subject</Var>
```

```
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“bool”>true</Ind>
            </rhs>
          </Equal>
          <Equal>
            <lhs>
              <Atom>
                <Rel>EWO_Valid_Query</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“bool”>true</Ind>
            </rhs>
          </Equal>
        </Or>
      </if>
      <do>
        <Atom>
          <Rel>Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert SD Secret IB 2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule16</oid>
      <!--Alert SD Secret IB 2-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Alert Exists (SECRET IB) for Port of
San Diego”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Unclassified”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Harbormaster”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“San Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 3</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>2/26/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule17</oid>
      <!--IB Results 3-->
      <if>
        <And>
          <And>
            <Atom>
              <Rel>Valid_Query</Rel>
              <Var>Subject</Var>
            </Atom>
```

```
            <Equal>
              <lhs>
                <Atom>
                  <Rel>IB_Response_Wait</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“bool”>false</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Security_Level</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Secret”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Secret IB and
Unclassified IB”</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Oakland Harbormaster query is valid
origination</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
```

```
            <Expr>
              <Fun
                uri=“dc:date”>
                <Ind>3/5/2009</Ind>
              </Fun>
            </Expr>
          </Plex>
        </label>
        <scope>
          <Ind
            uri=“#Information_Declassifier” />
        </scope>
        <oid>rule18</oid>
        <!--Oakland Harbormaster query is valid origination-->
        <if>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Harbormaster Oakland”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>HM_Queries</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Originating_Port Query”</Ind>
                </rhs>
              </Equal>
            </And>
            <Atom>
              <Rel>PDP_Decision</Rel>
              <Var>Subject</Var>
            </Atom>
          </And>
        </if>
        <do>
          <Atom>
            <Rel>HM_Valid_Query</Rel>
            <Var>Subject</Var>
            <Ind
              type=“bool”>true</Ind>
          </Atom>
```

```
    <Atom>
      <Rel>User_Role_Location</Rel>
      <Var>Subject</Var>
      <Ind
        type=“string”>“Oakland”</Ind>
    </Atom>
    <Atom>
      <Rel>Vessel_Classification_Level</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
    <Atom>
      <Rel>Vessel_Originating_Port</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Role_Location</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
  </do>
</Rule>
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>Alert SD TS IB</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:date”>
          <Ind>3/5/2009</Ind>
        </Fun>
      </Expr>
    </Plex>
  </label>
  <scope>
    <Ind
      uri=“#Information_Declassifier” />
  </scope>
  <oid>rule19</oid>
  <!--Alert SD TS IB-->
```

```
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (TS IB) for Port of San
Diego”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Top Secret”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Harbormaster San Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
```

```
        <label>
          <Plex>
            <Expr>
              <Fun
                uri=“dc:title”>
                <Ind>EWO query is valid 2</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:author”>
                <Ind>Randy Arvay</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:date”>
                <Ind>3/5/2009</Ind>
              </Fun>
            </Expr>
          </Plex>
        </label>
        <scope>
          <Ind
            uri=“#Information_Declassifier” />
        </scope>
        <oid>rule20</oid>
        <!--EWO query is valid 2-->
        <if>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Electronic Warfare Officer”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Top Secret”</Ind>
                </rhs>
              </Equal>
            </And>
            <Atom>
              <Rel>PDP_Decision</Rel>
              <Var>Subject</Var>
            </Atom>
          </And>
        </if>
        <do>
          <Atom>
            <Rel>EWO_Valid_Query</Rel>
            <Var>Subject</Var>
            <Ind
              type=“bool”>true</Ind>
          </Atom>
          <Atom>
            <Rel>Vessel_Classification_Level</Rel>
            <Var>Subject</Var>
            <Atom>
              <Rel>User_Security_Level</Rel>
              <Var>Subject</Var>
            </Atom>
          </Atom>
        </do>
      </Rule>
      <Rule
        style=“active”
        evaluation=“strong”>
        <label>
          <Plex>
            <Expr>
              <Fun
                uri=“dc:title”>
                <Ind>IB Results 4</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:author”>
                <Ind>Randy Arvay</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:date”>
                <Ind>2/26/2009</Ind>
              </Fun>
            </Expr>
          </Plex>
        </label>
        <scope>
          <Ind
            uri=“#Information_Declassifier” />
        </scope>
        <oid>rule21</oid>
        <!--IB Results 4-->
        <if>
          <And>
            <And>
              <Atom>
                <Rel>Valid_Query</Rel>
                <Var>Subject</Var>
              </Atom>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>IB_Response_Wait</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“bool”>false</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
        </if>
        <do>
          <Atom>
            <Rel>Return_IB_Results</Rel>
            <Var>Subject</Var>
            <Ind
              type=“string”>“Vessel Results from Unclassified IB”</Ind>
          </Atom>
        </do>
      </Rule>
      <Rule
        style=“active”
        evaluation=“strong”>
        <label>
          <Plex>
            <Expr>
              <Fun
                uri=“dc:title”>
                <Ind>Oakland Harbormaster query is valid origination
  2</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:author”>
                <Ind>Randy Arvay</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:date”>
                <Ind>3/5/2009</Ind>
              </Fun>
            </Expr>
          </Plex>
        </label>
        <scope>
          <Ind
            uri=“#Information_Declassifier” />
        </scope>
        <oid>rule22</oid>
        <!--Oakland Harbormaster query is valid origination 2-->
        <if>
          <And>
            <And>
              <And>
                <Equal>
                  <lhs>
                    <Atom>
                      <Rel>User_Role</Rel>
                      <Var>Subject</Var>
                    </Atom>
                  </lhs>
                  <rhs>
                    <Ind
                      type=“string”>“Harbormaster”</Ind>
                  </rhs>
                </Equal>
                <Equal>
                  <lhs>
                    <Atom>
                      <Rel>User_Role_Location</Rel>
                      <Var>Subject</Var>
                    </Atom>
                  </lhs>
                  <rhs>
                    <Ind
                      type=“string”>“Oakland”</Ind>
                  </rhs>
                </Equal>
              </And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>HM_Queries</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Originating_Port Query”</Ind>
                </rhs>
              </Equal>
            </And>
            <Atom>
              <Rel>PDP_Decision</Rel>
              <Var>Subject</Var>
            </Atom>
          </And>
        </if>
        <do>
          <Atom>
            <Rel>HM_Valid_Query</Rel>
            <Var>Subject</Var>
            <Ind
              type=“bool”>true</Ind>
          </Atom>
          <Atom>
            <Rel>Vessel_Classification_Level</Rel>
            <Var>Subject</Var>
            <Atom>
              <Rel>User_Security_Level</Rel>
              <Var>Subject</Var>
            </Atom>
          </Atom>
          <Atom>
            <Rel>Vessel_Originating_Port</Rel>
            <Var>Subject</Var>
            <Atom>
              <Rel>User_Role_Location</Rel>
              <Var>Subject</Var>
            </Atom>
          </Atom>
        </do>
      </Rule>
      <Rule
        style=“active”
        evaluation=“strong”>
        <label>
          <Plex>
            <Expr>
              <Fun
                uri=“dc:title”>
                <Ind>Alert SD TS IB 2</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:author”>
                <Ind>Randy Arvay</Ind>
              </Fun>
            </Expr>
```

```
          <Expr>
            <Fun
              uri="dc:date">
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri="#Information_Declassifier" />
      </scope>
      <oid>rule23</oid>
      <!--Alert SD TS IB 2-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Alert Exists (TS IB) for Port of San
Diego"</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type="string">"Top Secret"</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Harbormaster"</Ind>
              </rhs>
```

```
                </Equal>
              </And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role_Location</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“San Diego”</Ind>
                </rhs>
              </Equal>
            </And>
          </if>
          <do>
            <Atom>
              <Rel>Alert_Present_Response</Rel>
              <Var>Subject</Var>
              <Ind
                type=“bool”>true</Ind>
            </Atom>
          </do>
        </Rule>
        <Rule
          style=“active”
          evaluation=“strong”>
          <label>
            <Plex>
              <Expr>
                <Fun
                  uri=“dc:title”>
                  <Ind>IB Results 5</Ind>
                </Fun>
              </Expr>
              <Expr>
                <Fun
                  uri=“dc:author”>
                  <Ind>Randy Arvay</Ind>
                </Fun>
              </Expr>
              <Expr>
                <Fun
                  uri=“dc:date”>
                  <Ind>3/5/2009</Ind>
                </Fun>
              </Expr>
            </Plex>
          </label>
          <scope>
            <Ind
              uri=“#Information_Declassifier” />
          </scope>
          <oid>rule24</oid>
```

```
   <!--IB Results 5-->
   <if>
    <And>
     <And>
      <Atom>
       <Rel>Valid_Query</Rel>
       <Var>Subject</Var>
      </Atom>
      <Equal>
       <lhs>
        <Atom>
         <Rel>IB_Response_Wait</Rel>
         <Var>Subject</Var>
        </Atom>
       </lhs>
       <rhs>
        <Ind
         type=“bool”>false</Ind>
       </rhs>
      </Equal>
     </And>
     <Equal>
      <lhs>
       <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
       </Atom>
      </lhs>
      <rhs>
       <Ind
        type=“string”>“Top Secret”</Ind>
      </rhs>
     </Equal>
    </And>
   </if>
   <do>
    <Atom>
     <Rel>Return_IB_Results</Rel>
     <Var>Subject</Var>
     <Ind
      type=“string”>“Vessel Results from TS IB, Secret IB, and
Unclassified IB”</Ind>
    </Atom>
   </do>
  </Rule>
  <Rule
   style=“active”
   evaluation=“strong”>
   <label>
    <Plex>
     <Expr>
      <Fun
       uri=“dc:title”>
       <Ind>SD Harbormaster query is valid destination</Ind>
      </Fun>
    </Expr>
    <Expr>
      <Fun
        uri=“dc:author”>
        <Ind>Randy Arvay</Ind>
      </Fun>
    </Expr>
    <Expr>
      <Fun
        uri=“dc:date”>
        <Ind>3/5/2009</Ind>
      </Fun>
    </Expr>
  </Plex>
</label>
<scope>
  <Ind
    uri=“#Information_Declassifier” />
</scope>
<oid>rule25</oid>
<!--SD Harbormaster query is valid destination-->
<if>
  <And>
    <And>
      <And>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Role</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“Harbormaster”</Ind>
          </rhs>
        </Equal>
        <Equal>
          <lhs>
            <Atom>
              <Rel>User_Role_Location</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“San Diego”</Ind>
          </rhs>
        </Equal>
      </And>
      <Equal>
        <lhs>
          <Atom>
            <Rel>HM_Queries</Rel>
            <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“Destination_Port Query”</Ind>
          </rhs>
        </Equal>
      </And>
      <Atom>
        <Rel>PDP_Decision</Rel>
        <Var>Subject</Var>
      </Atom>
    </And>
  </if>
  <do>
    <Atom>
      <Rel>HM_Valid_Query</Rel>
      <Var>Subject</Var>
      <Ind
        type=“bool”>true</Ind>
    </Atom>
    <Atom>
      <Rel>Vessel_Classification_Level</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Security_Level</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
    <Atom>
      <Rel>Vessel_Destination_Port</Rel>
      <Var>Subject</Var>
      <Atom>
        <Rel>User_Role_Location</Rel>
        <Var>Subject</Var>
      </Atom>
    </Atom>
  </do>
</Rule>
<Rule
  style=“active”
  evaluation=“strong”>
  <label>
    <Plex>
      <Expr>
        <Fun
          uri=“dc:title”>
          <Ind>Alert Oak Secret IB</Ind>
        </Fun>
      </Expr>
      <Expr>
        <Fun
          uri=“dc:author”>
          <Ind>Randy Arvay</Ind>
        </Fun>
```

```
          </Expr>
          <Expr>
            <Fun
              uri="dc:date">
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri="#Information_Declassifier" />
      </scope>
      <oid>rule26</oid>
      <!--Alert Oak Secret IB-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Alert Exists (SECRET IB) for Port of
Oakland"</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type="string">"Unclassified"</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type="string">"Harbormaster Oakland"</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 6</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule27</oid>
      <!--IB Results 6-->
      <if>
        <Equal>
          <lhs>
            <Atom>
              <Rel>Valid_Query</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“bool”>false</Ind>
          </rhs>
        </Equal>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Invalid Query!”</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>SD Harbormaster query is valid destination 2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule28</oid>
      <!--SD Harbormaster query is valid destination 2-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Harbormaster San Diego”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>HM_Queries</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Destination_Port Query”</Ind>
              </rhs>
            </Equal>
          </And>
          <Atom>
            <Rel>PDP_Decision</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
        <Atom>
          <Rel>User_Role_Location</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“San Diego”</Ind>
        </Atom>
        <Atom>
          <Rel>Vessel_Classification_Level</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Security_Level</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
        <Atom>
          <Rel>Vessel_Destination_Port</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Role_Location</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
```

```
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert Oak Secret IB 2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule29</oid>
      <!--Alert Oak Secret IB 2-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Alert Exists (SECRET IB) for Port of
Oakland”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Unclassified”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Harbormaster”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Oakland”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>SD Harbormaster query is valid origination</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule30</oid>
      <!--SD Harbormaster query is valid origination-->
      <if>
        <And>
          <And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Harbormaster San Diego”</Ind>
              </rhs>
            </Equal>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>HM_Queries</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Originating_Port Query”</Ind>
              </rhs>
            </Equal>
          </And>
          <Atom>
            <Rel>PDP_Decision</Rel>
            <Var>Subject</Var>
          </Atom>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
        <Atom>
          <Rel>User_Role_Location</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“San Diego”</Ind>
        </Atom>
        <Atom>
          <Rel>Vessel_Classification_Level</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Security_Level</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
        <Atom>
          <Rel>Vessel_Originating_Port</Rel>
          <Var>Subject</Var>
          <Atom>
            <Rel>User_Role_Location</Rel>
            <Var>Subject</Var>
          </Atom>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>Alert OakTS IB</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule31</oid>
```

```
      <!--Alert OakTS IB-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Alert Exists (TS IB) for Port of
Oakland”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Top Secret”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Harbormaster”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Oakland”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>true</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>SD Harbormaster query is valid origination 2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule32</oid>
      <!--SD Harbormaster query is valid origination 2-->
      <if>
        <And>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Role</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
```

```
                    <Ind
                      type=“string”>“Harbormaster”</Ind>
                  </rhs>
                </Equal>
                <Equal>
                  <lhs>
                    <Atom>
                      <Rel>User_Role_Location</Rel>
                      <Var>Subject</Var>
                    </Atom>
                  </lhs>
                  <rhs>
                    <Ind
                      type=“string”>“San Diego”</Ind>
                  </rhs>
                </Equal>
              </And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>HM_Queries</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Originating_Port Query”</Ind>
                </rhs>
              </Equal>
            </And>
            <Atom>
              <Rel>PDP_Decision</Rel>
              <Var>Subject</Var>
            </Atom>
          </And>
        </if>
        <do>
          <Atom>
            <Rel>HM_Valid_Query</Rel>
            <Var>Subject</Var>
            <Ind
              type=“bool”>true</Ind>
          </Atom>
          <Atom>
            <Rel>Vessel_Classification_Level</Rel>
            <Var>Subject</Var>
            <Atom>
              <Rel>User_Security_Level</Rel>
              <Var>Subject</Var>
            </Atom>
          </Atom>
          <Atom>
            <Rel>Vessel_Originating_Port</Rel>
            <Var>Subject</Var>
            <Atom>
              <Rel>User_Role_Location</Rel>
              <Var>Subject</Var>
            </Atom>
          </Atom>
        </do>
      </Rule>
      <Rule
        style=“active”
        evaluation=“strong”>
        <label>
          <Plex>
            <Expr>
              <Fun
                uri=“dc:title”>
                <Ind>Alert OakTS IB 2</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:author”>
                <Ind>Randy Arvay</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:date”>
                <Ind>3/5/2009</Ind>
              </Fun>
            </Expr>
          </Plex>
        </label>
        <scope>
          <Ind
            uri=“#Information_Declassifier” />
        </scope>
        <oid>rule33</oid>
        <!--Alert OakTS IB 2-->
        <if>
          <And>
            <And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>Alert_Notification</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Alert Exists (TS IB) for Port of
  Oakland”</Ind>
                </rhs>
              </Equal>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Top Secret”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Role</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Harbormaster Oakland”</Ind>
              </rhs>
            </Equal>
          </And>
        </if>
        <do>
          <Atom>
            <Rel>Alert_Present_Response</Rel>
            <Var>Subject</Var>
            <Ind
              type=“bool”>true</Ind>
          </Atom>
        </do>
      </Rule>
      <Rule
        style=“active”
        evaluation=“strong”>
        <label>
          <Plex>
            <Expr>
              <Fun
                uri=“dc:title”>
                <Ind>IB Results 0.1</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:author”>
                <Ind>Randy Arvay</Ind>
              </Fun>
            </Expr>
            <Expr>
              <Fun
                uri=“dc:date”>
                <Ind>3/5/2009</Ind>
```

```
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule34</oid>
      <!--IB Results 0.1-->
      <if>
        <And>
          <And>
            <And>
              <Atom>
                <Rel>Valid_Query</Rel>
                <Var>Subject</Var>
              </Atom>
              <Atom>
                <Rel>IB_Response_Wait</Rel>
                <Var>Subject</Var>
              </Atom>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Notification</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Alert Exists (SECRET IB) for Port of
Oakland”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Unclassified IB with ID
Injection for Oakland Alert”</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 0.2</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule35</oid>
      <!--IB Results 0.2-->
      <if>
        <And>
          <And>
            <And>
              <Atom>
                <Rel>Valid_Query</Rel>
                <Var>Subject</Var>
              </Atom>
              <Atom>
                <Rel>IB_Response_Wait</Rel>
                <Var>Subject</Var>
              </Atom>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Notification</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Alert Exists (SECRET IB) for Port of San
Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Unclassified IB with ID
Injection for San Diego Alert”</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 0.3</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule36</oid>
      <!--IB Results 0.3-->
      <if>
        <And>
          <And>
            <And>
              <Atom>
                <Rel>Valid_Query</Rel>
                <Var>Subject</Var>
              </Atom>
              <Atom>
                <Rel>IB_Response_Wait</Rel>
                <Var>Subject</Var>
              </Atom>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>User_Security_Level</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Unclassified”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>Alert_Notification</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Alert Exists (TS IB) for Port of San
Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
```

```
            type=“string”>“Vessel Results from Unclassified IB with ID
Injection for San Diego Alert”</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>No Alert IB</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/10/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule37</oid>
      <!--No Alert IB-->
      <if>
        <Equal>
          <lhs>
            <Atom>
              <Rel>Alert_Notification</Rel>
              <Var>Subject</Var>
            </Atom>
          </lhs>
          <rhs>
            <Ind
              type=“string”>“No Alert Exists”</Ind>
          </rhs>
        </Equal>
      </if>
      <do>
        <Atom>
          <Rel>Alert_Present_Response</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>IB Results 2.1</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/5/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule38</oid>
      <!--IB Results 2.1-->
      <if>
        <And>
          <And>
            <And>
              <And>
                <Atom>
                  <Rel>Valid_Query</Rel>
                  <Var>Subject</Var>
                </Atom>
                <Atom>
                  <Rel>IB_Response_Wait</Rel>
                  <Var>Subject</Var>
                </Atom>
              </And>
              <Equal>
                <lhs>
                  <Atom>
                    <Rel>User_Security_Level</Rel>
                    <Var>Subject</Var>
                  </Atom>
```

```
                </lhs>
                <rhs>
                  <Ind
                    type=“string”>“Secret”</Ind>
                </rhs>
              </Equal>
            </And>
            <Equal>
              <lhs>
                <Atom>
                  <Rel>Alert_Notification</Rel>
                  <Var>Subject</Var>
                </Atom>
              </lhs>
              <rhs>
                <Ind
                  type=“string”>“Alert Exists (TS IB) for Port of San
Diego”</Ind>
              </rhs>
            </Equal>
          </And>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role_Location</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“San Diego”</Ind>
            </rhs>
          </Equal>
        </And>
      </if>
      <do>
        <Atom>
          <Rel>Return_IB_Results</Rel>
          <Var>Subject</Var>
          <Ind
            type=“string”>“Vessel Results from Secret IB and
Unclassified IB with ID Injection for San Diego Alert”</Ind>
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>EWO query is invalid</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/10/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule39</oid>
      <!--EWO query is invalid-->
      <if>
        <Or>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Electronic Warfare Officer”</Ind>
            </rhs>
          </Equal>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“EWO”</Ind>
            </rhs>
          </Equal>
        </Or>
      </if>
      <do>
        <Atom>
          <Rel>EWO_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
```

```
        </Atom>
      </do>
    </Rule>
    <Rule
      style=“active”
      evaluation=“strong”>
      <label>
        <Plex>
          <Expr>
            <Fun
              uri=“dc:title”>
              <Ind>HM query is invalid</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:author”>
              <Ind>Randy Arvay</Ind>
            </Fun>
          </Expr>
          <Expr>
            <Fun
              uri=“dc:date”>
              <Ind>3/10/2009</Ind>
            </Fun>
          </Expr>
        </Plex>
      </label>
      <scope>
        <Ind
          uri=“#Information_Declassifier” />
      </scope>
      <oid>rule40</oid>
      <!--HM query is invalid-->
      <if>
        <Or>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Harbormaster Oakland”</Ind>
            </rhs>
          </Equal>
          <Equal>
            <lhs>
              <Atom>
                <Rel>User_Role</Rel>
                <Var>Subject</Var>
              </Atom>
            </lhs>
            <rhs>
              <Ind
                type=“string”>“Harbormaster San Diego”</Ind>
            </rhs>
          </Equal>
        </Or>
      </if>
      <do>
        <Atom>
          <Rel>HM_Valid_Query</Rel>
          <Var>Subject</Var>
          <Ind
            type=“bool”>false</Ind>
        </Atom>
      </do>
    </Rule>
  </Rulebase>
</RuleML>
```

THIS PAGE INTENTIONALLY LEFT BLANK

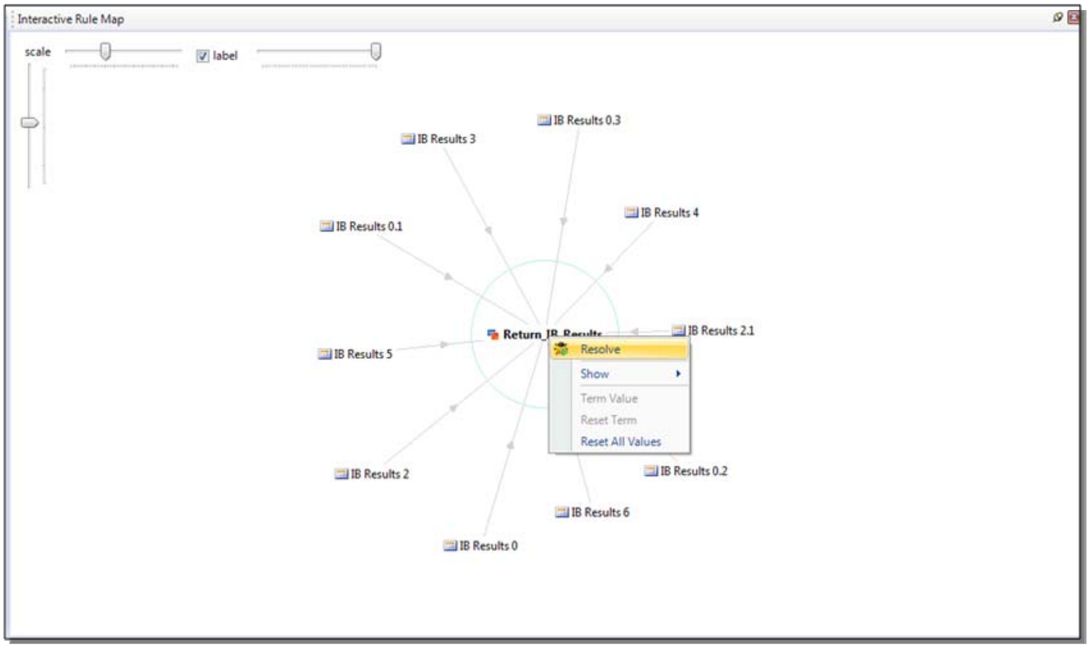
APPENDIX C. RULE TRACE OF USE CASE SUPPORTED

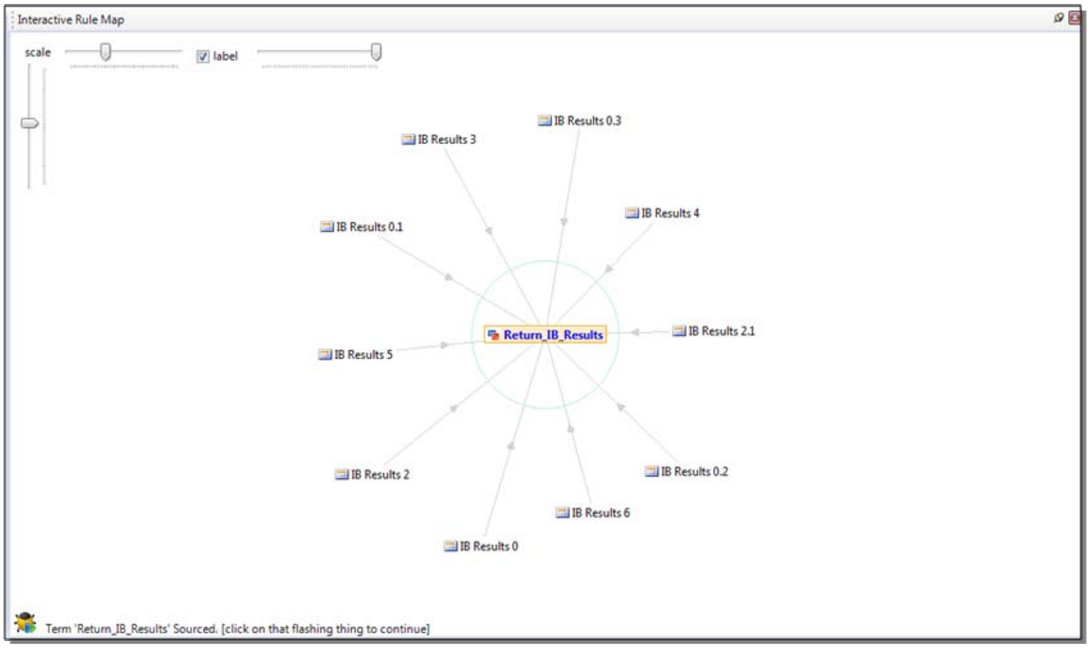
The visual trace of the ruleset execution to show support for the system's intended use. This trace is completed using the following parameters:

Type of query:	Destination Port
Role / Actor:	Harbormaster
Location:	San Diego
Security Level:	Unclassified
Alert Present:	False
Alert Classification Level:	Not Applicable
Expected Result:	"Vessel Results from Unclassified IB"

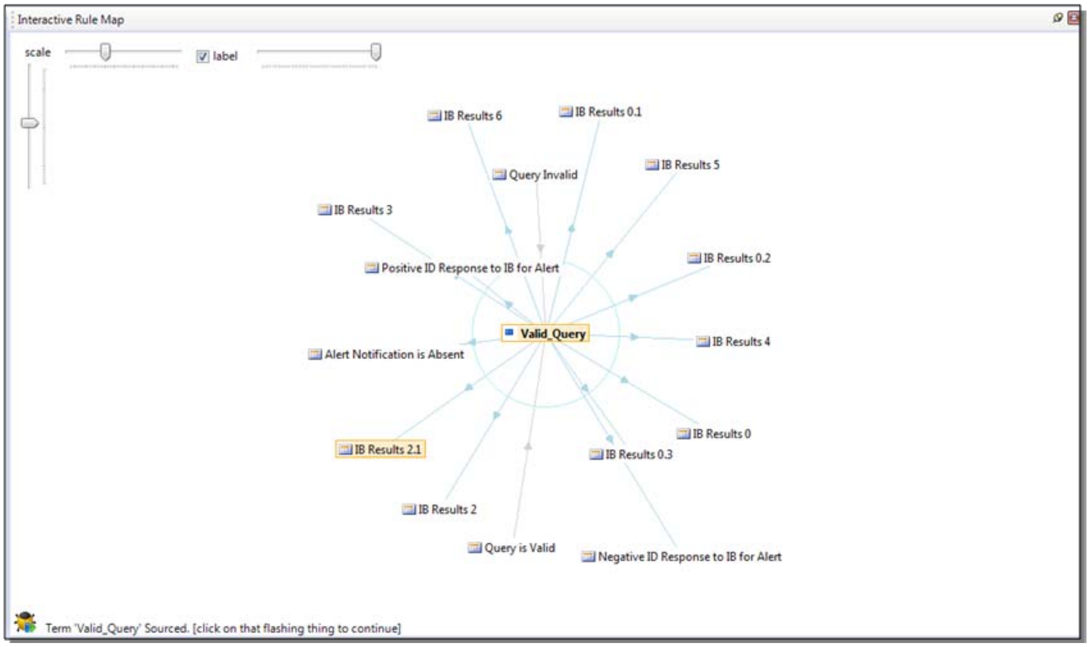
The expected result from this query and the RuleML execution is an IB response of "Vessel Results from Unclassified IB" to the user. The trace was completed using RuleManager's Interactive Rule Map functionality. The pop-up boxes shown throughout the trace indicate the sourcing of predicates that would be included with tagged data in a live system. This was not replicated for this research and instead was manually inserted via the dialog boxes.

For the rule trace shown and from the interactive rule map of RuleManager, various indicators are used to show the actions during the trace. A rule or variable highlighted in Yellow, indicates that a rule or variable is currently being sourced. A rule depicted in Red indicates that the rule did not fire (execute) from the ruleset. A rule shown highlighted in Green indicates that the rule did fire and the result of that firing is also shown in the green highlighted box with the predicate name. Pop-up dialog boxes shown in the trace indicate the sourcing of a variable or predicate that would be done by an external entity (service) or taken from XML attributes attached to the query upon origination (i.e., user role and user security level).

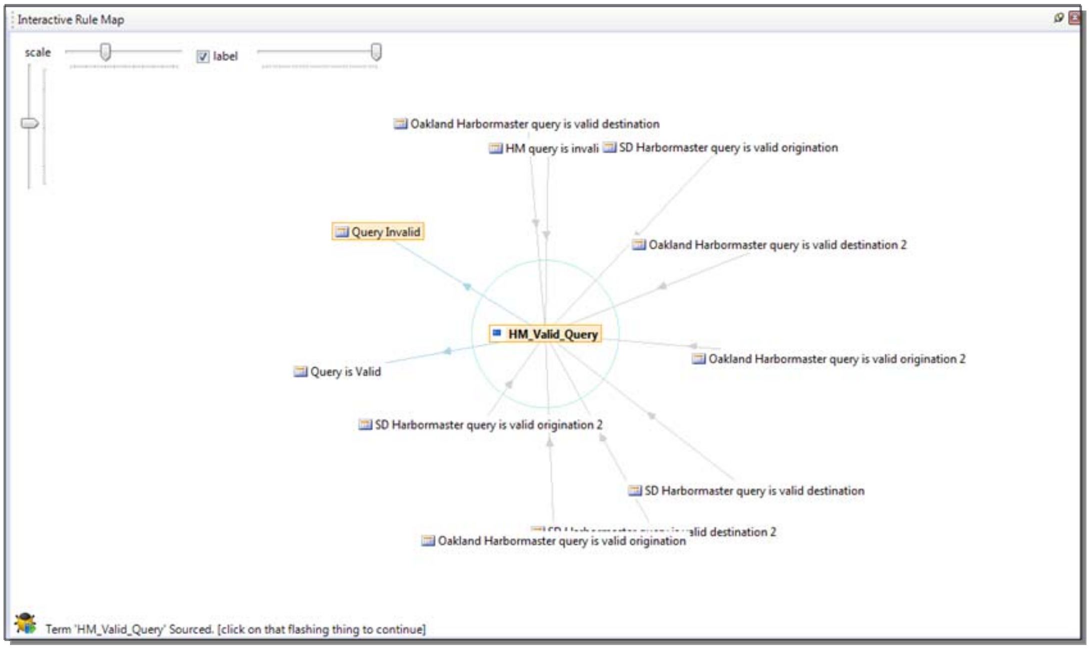




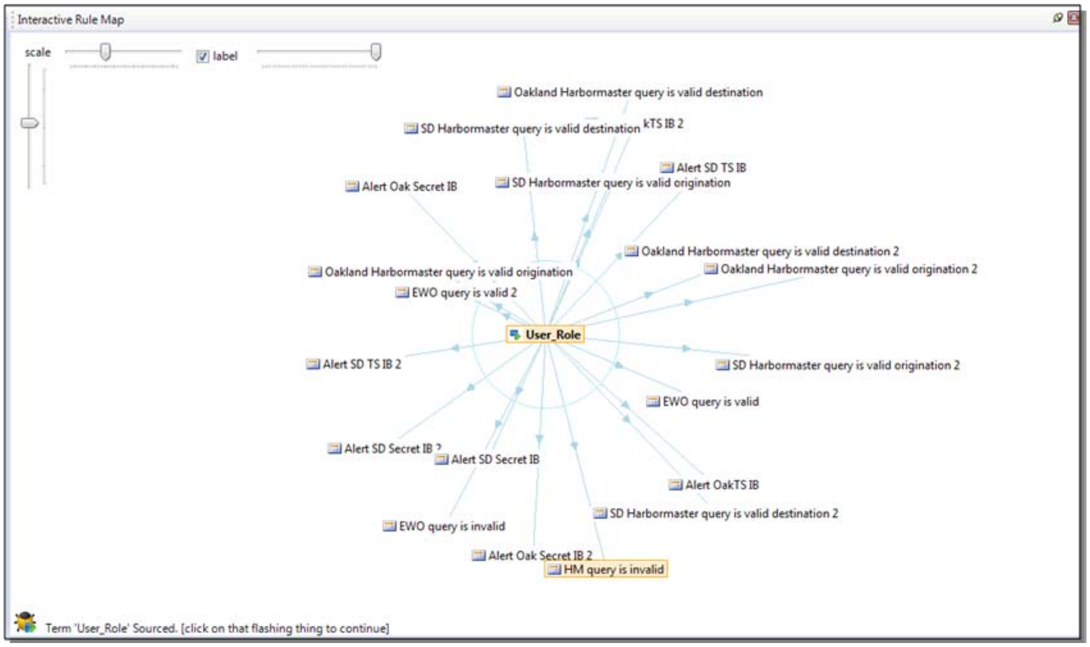
Term 'Return_IB_Results' Sourced. [click on that flashing thing to continue]



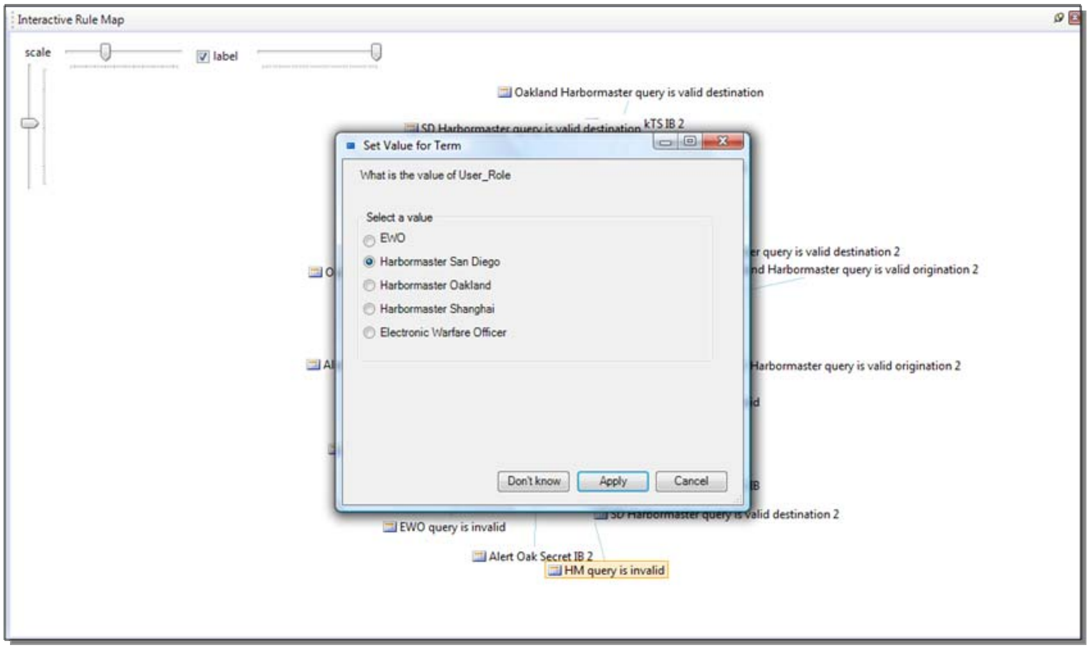
Term 'Valid_Query' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
HM query is invali
SD Harbormaster query is valid origination
Query Invalid
Oakland Harbormaster query is valid destination 2
HM_Valid_Query
Oakland Harbormaster query is valid origination 2
Query is Valid
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
alid destination 2
Oakland Harbormaster query is valid origination

Term 'HM_Valid_Query' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
kTS IB 2
Alert SD TS IB
Alert Oak Secret IB
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination
EWO query is valid 2
User_Role
Alert SD TS IB 2
SD Harbormaster query is valid origination 2
EWO query is valid
Alert SD Secret IB 2
Alert SD Secret IB
Alert OakTS IB
SD Harbormaster query is valid destination 2
EWO query is invalid
Alert Oak Secret IB 2
HM query is invalid

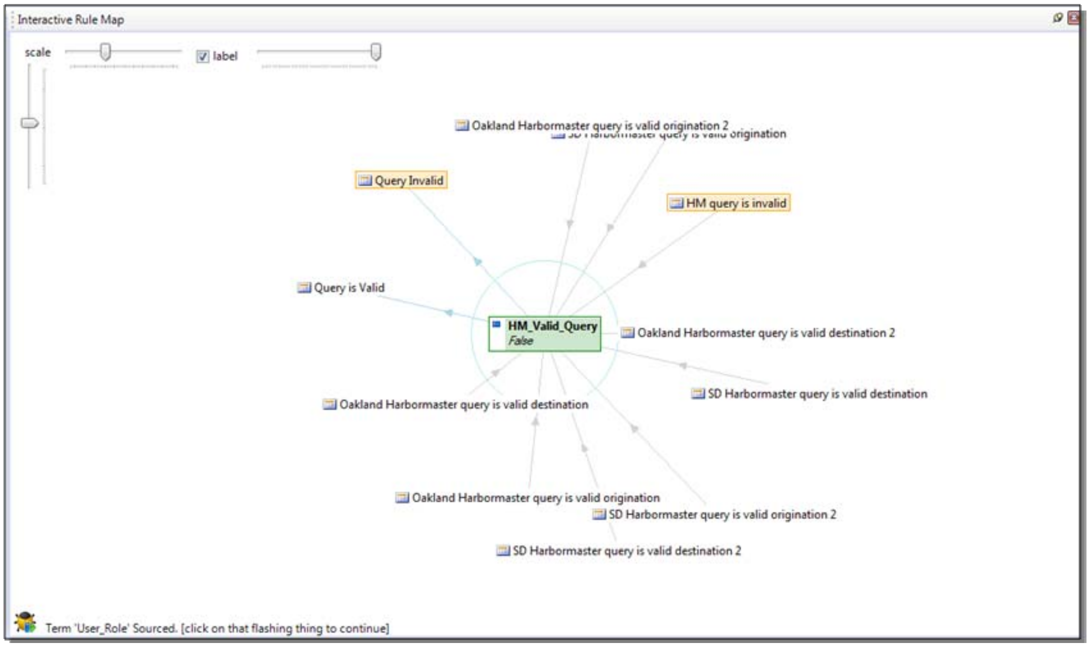
Term 'User_Role' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
kTS IB 2
er query is valid destination 2
nd Harbormaster query is valid origination 2
Harbormaster query is valid origination 2
SD Harbormaster query is valid destination 2
EWO query is invalid
Alert Oak Secret IB 2
HM query is invalid
Set Value for Term
Don't know
Apply
Cancel

What is the value of User_Role

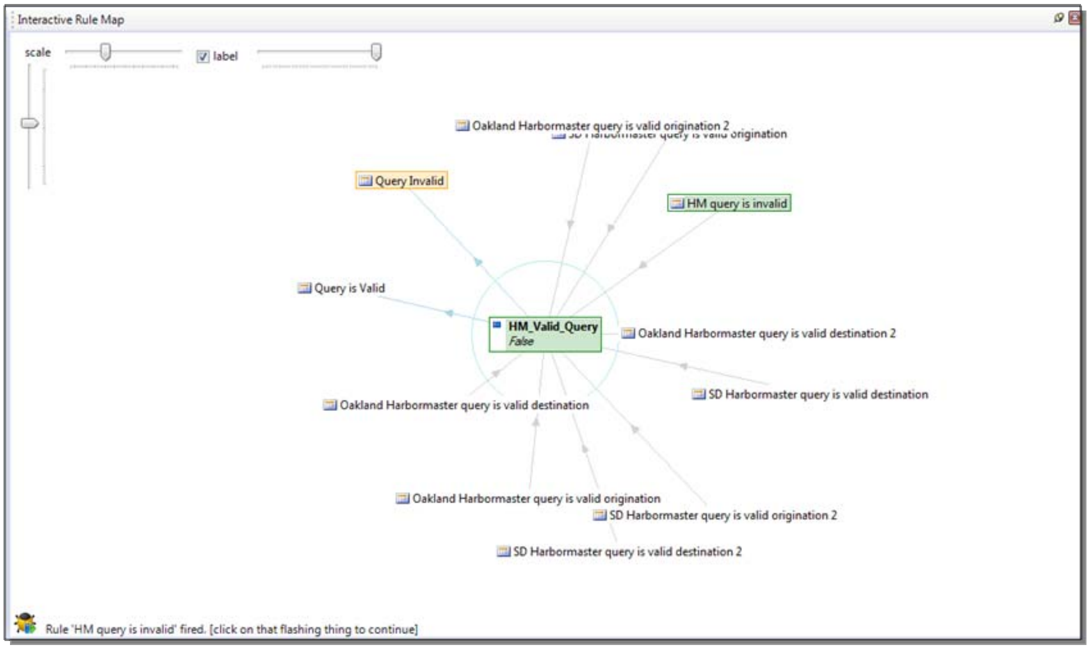
Select a value

○ EWO
● Harbormaster San Diego
○ Harbormaster Oakland
○ Harbormaster Shanghai
○ Electronic Warfare Officer



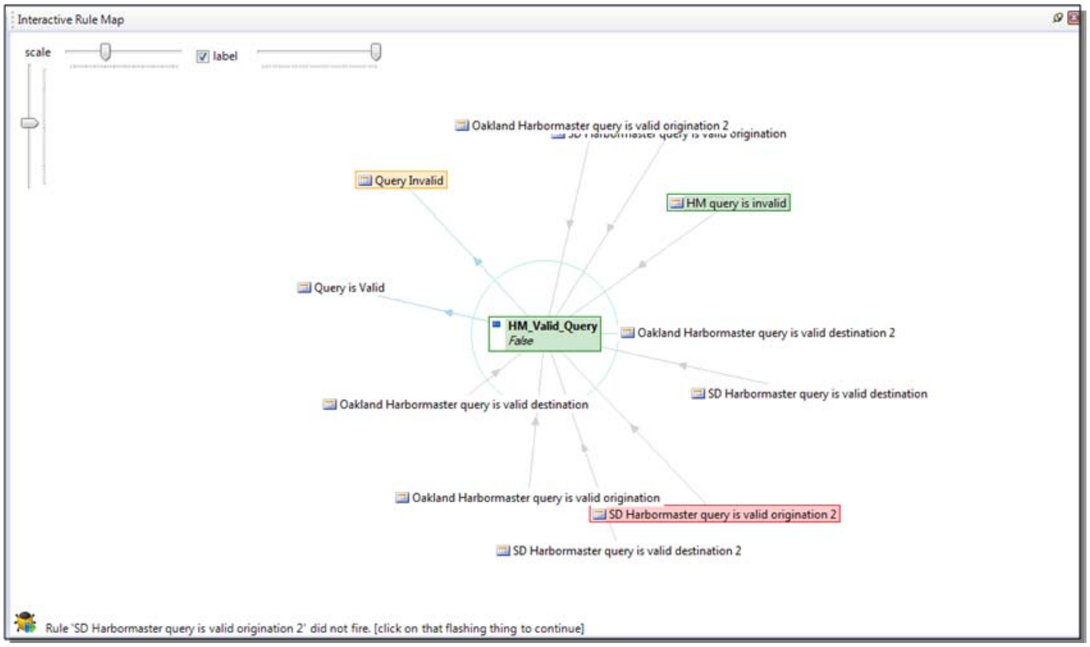
Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination
Query Invalid
HM query is invalid
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid destination
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination 2

Term 'User_Role' Sourced. [click on that flashing thing to continue]

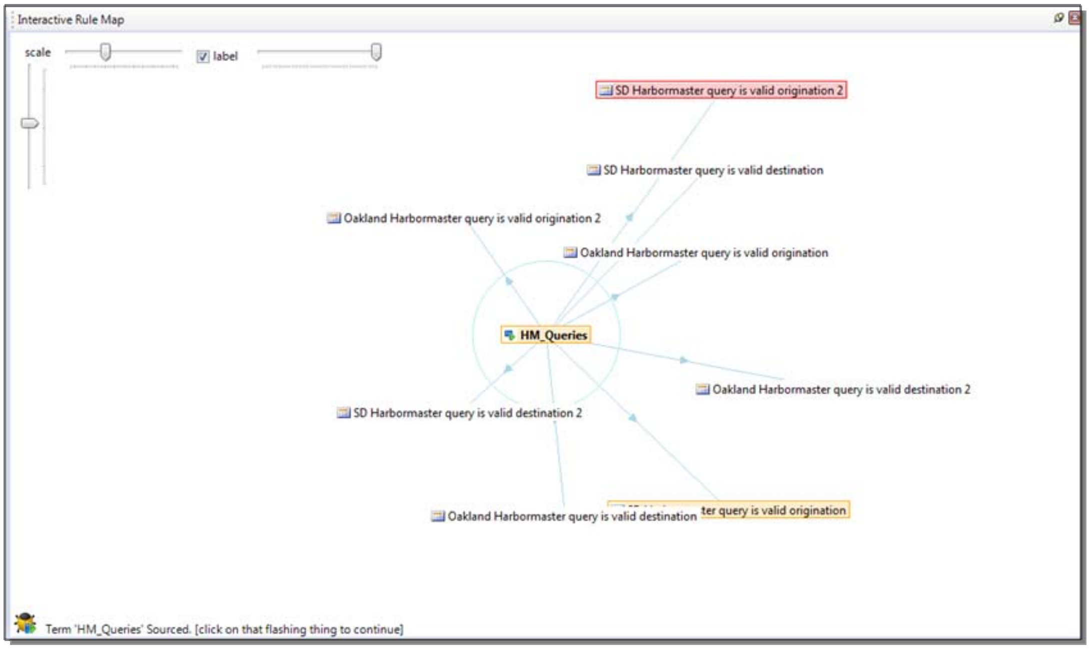


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination
Query Invalid
HM query is invalid
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid destination
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination 2

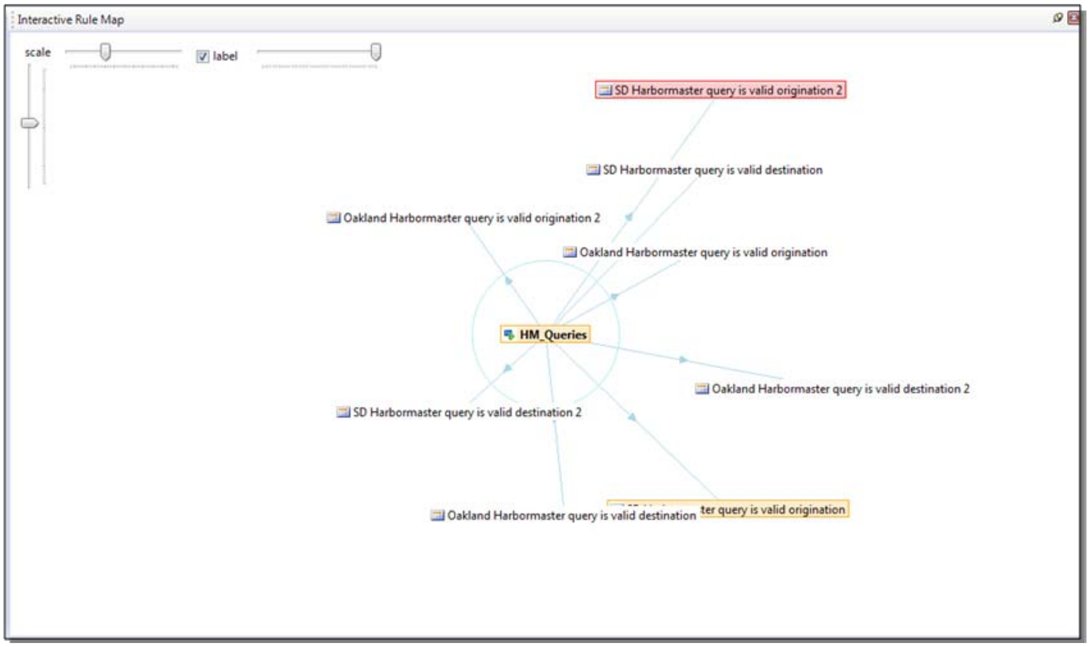
Rule 'HM query is invalid' fired. [click on that flashing thing to continue]

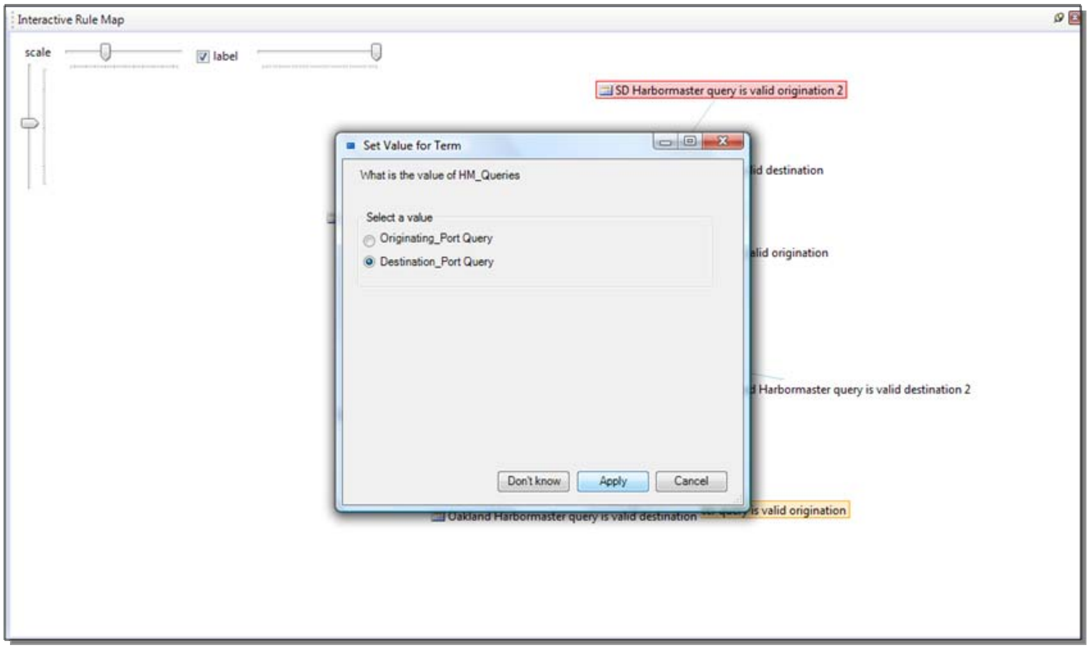

Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination
Query Invalid
HM query is invalid
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid destination
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination 2

Rule 'SD Harbormaster query is valid origination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination
HM_Queries
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
ter query is valid origination

Term 'HM_Queries' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination
HM_Queries
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
ter query is valid origination

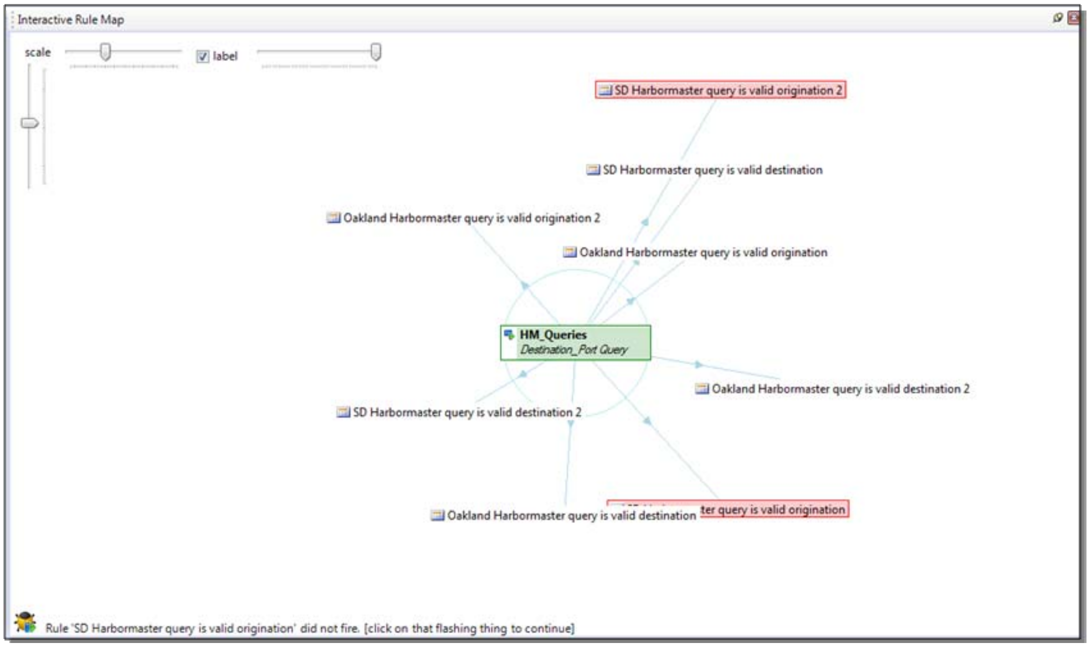

Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination 2
lid destination
alid origination
l Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
is valid origination
Set Value for Term

What is the value of HM_Queries

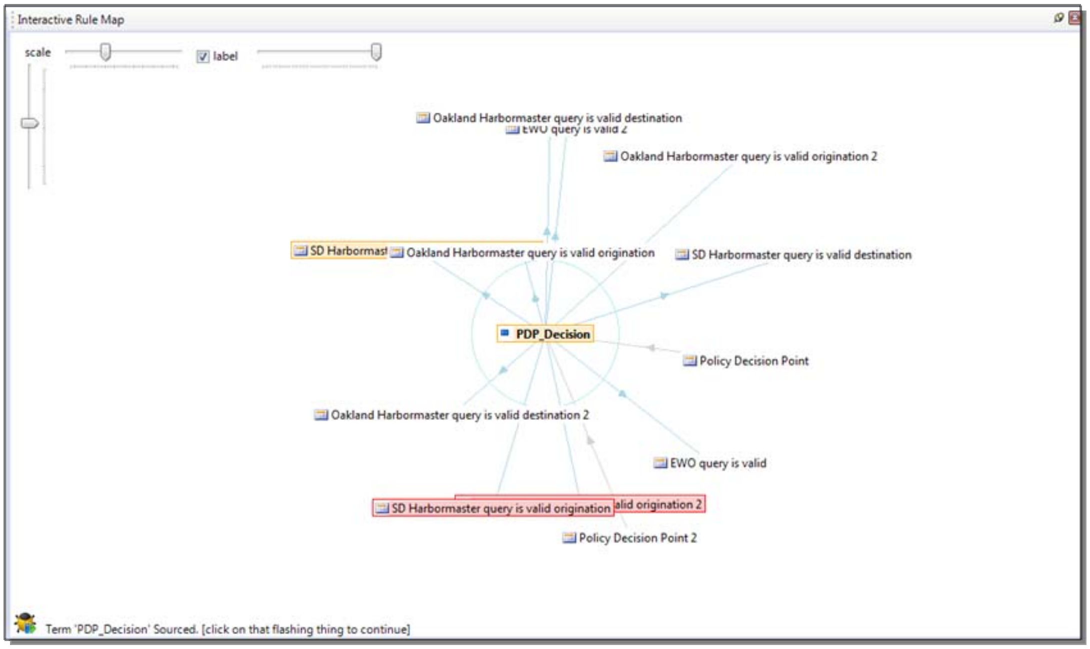
Select a value

○ Originating_Port Query

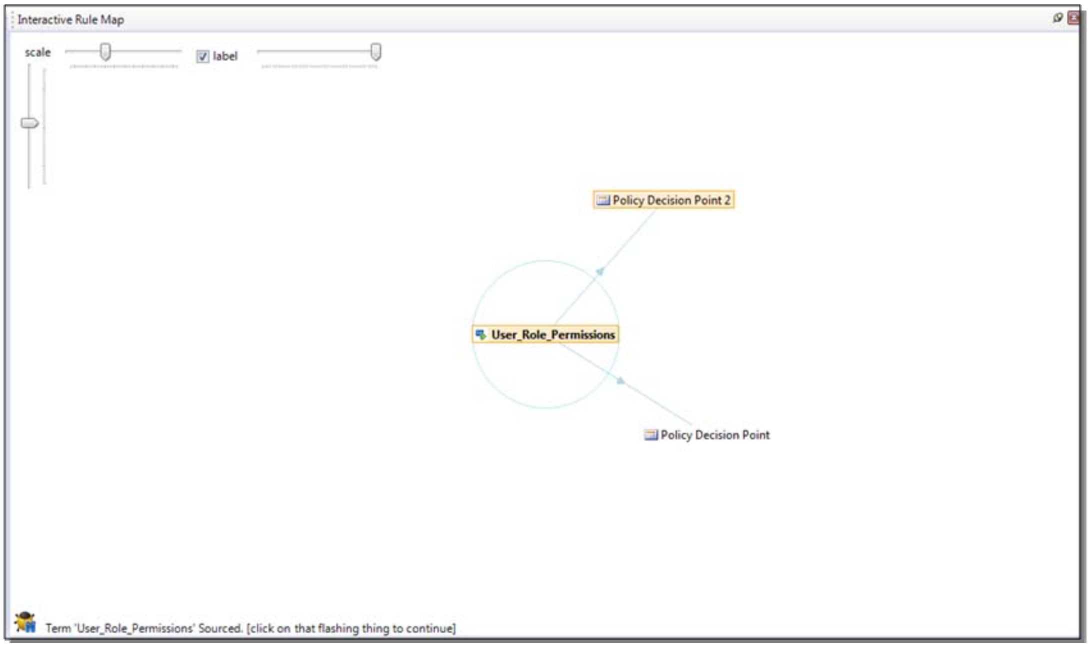
● Destination_Port Query


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination
HM_Queries
Destination_Port Query
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
ter query is valid origination

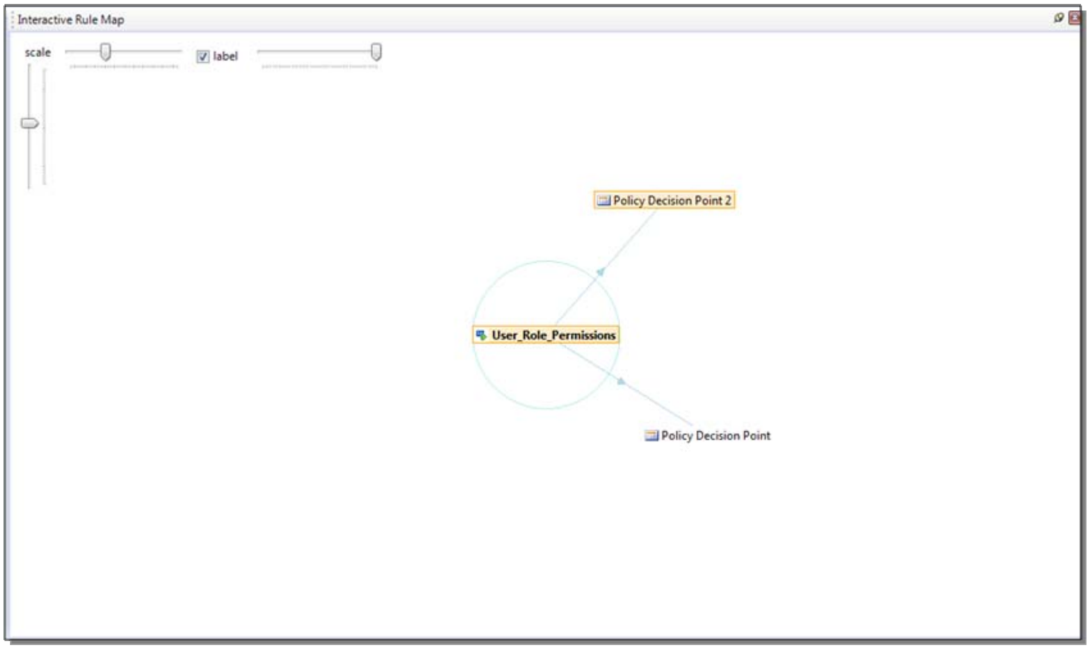
Rule 'SD Harbormaster query is valid origination' did not fire. [click on that flashing thing to continue]

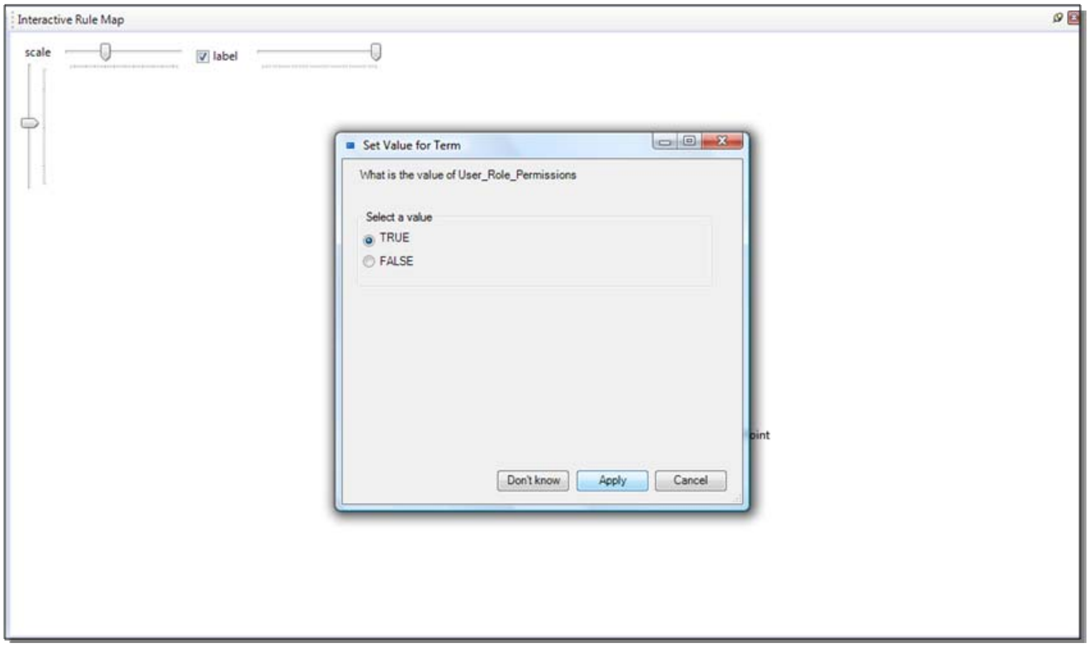

Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
EWO query is valid 2
Oakland Harbormaster query is valid origination 2
SD Harbormast
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
PDP_Decision
Policy Decision Point
Oakland Harbormaster query is valid destination 2
EWO query is valid
SD Harbormaster query is valid origination
alid origination 2
Policy Decision Point 2

Term 'PDP_Decision' Sourced. [click on that flashing thing to continue]



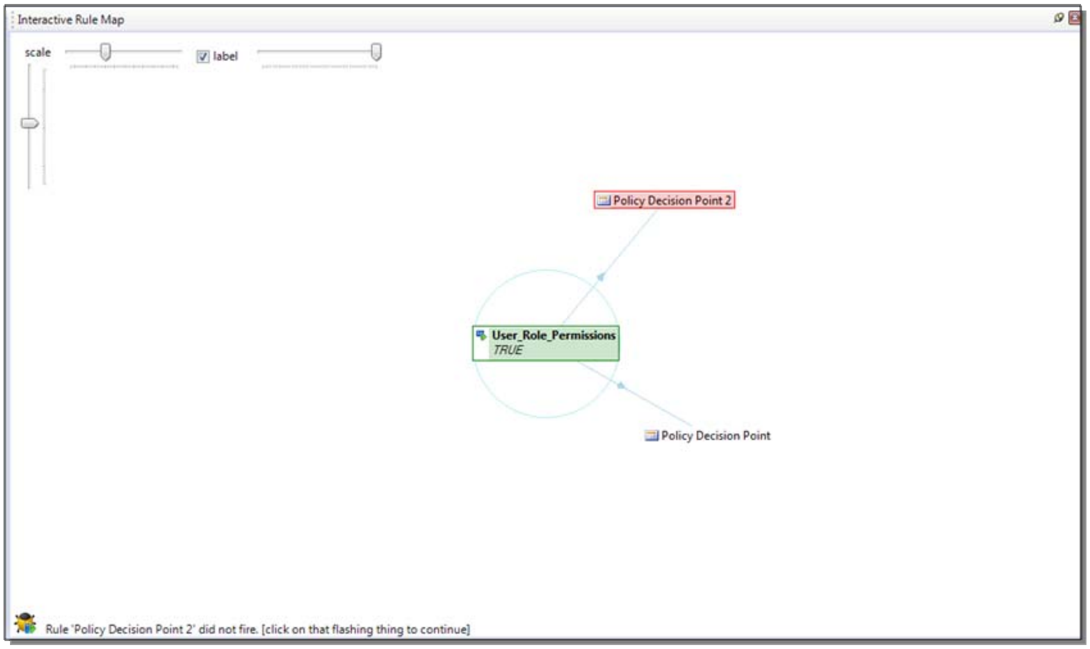
Term 'User_Role_Permissions' Sourced. [click on that flashing thing to continue]



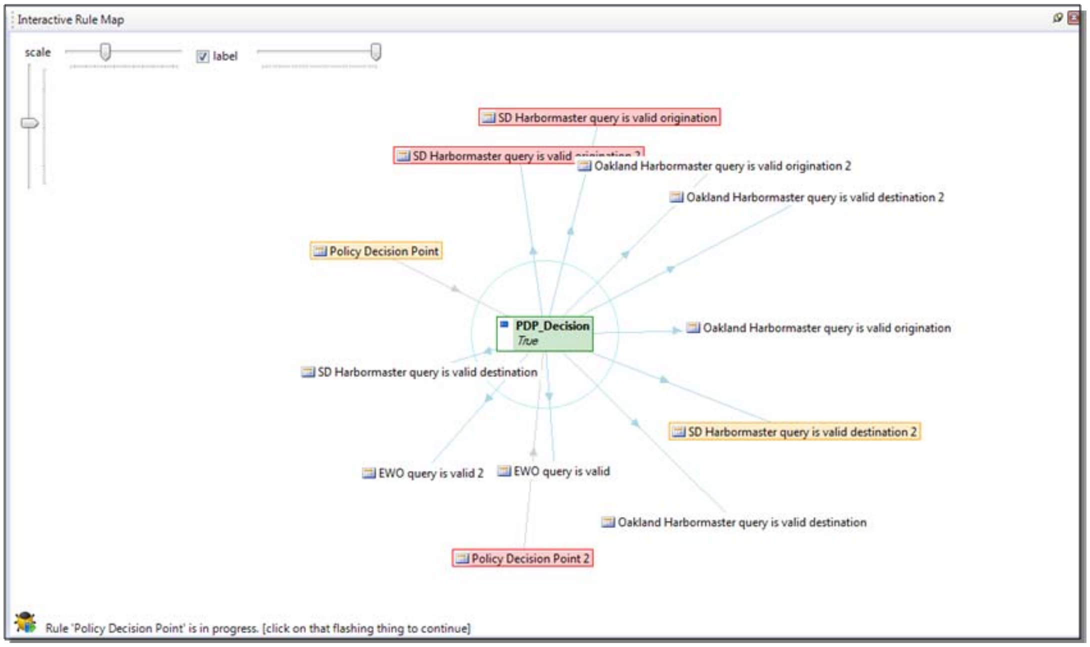


What is the value of User_Role_Permissions

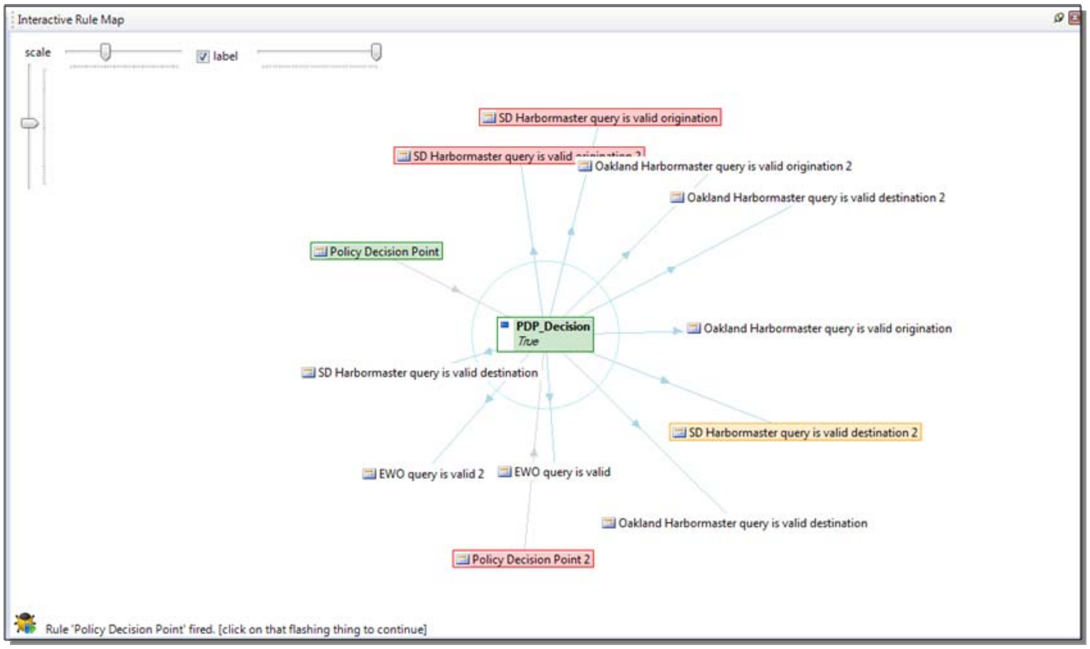
Select a value
● TRUE
○ FALSE



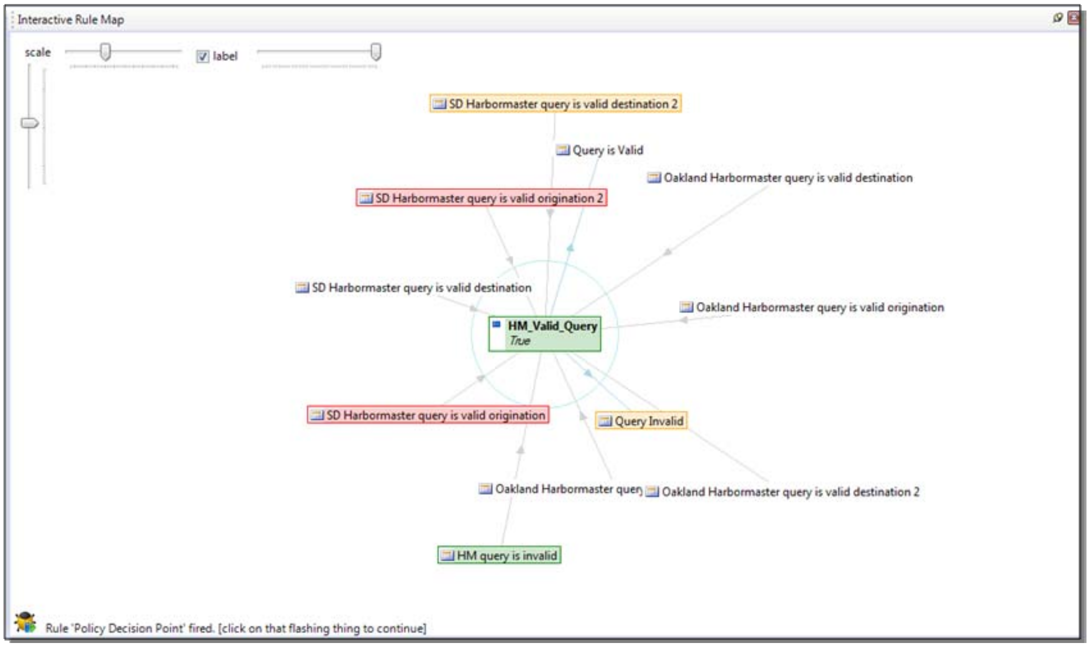
Rule 'Policy Decision Point 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination 2
Policy Decision Point
PDP_Decision
True
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
SD Harbormaster query is valid destination 2
EWO query is valid 2
EWO query is valid
Oakland Harbormaster query is valid destination
Policy Decision Point 2

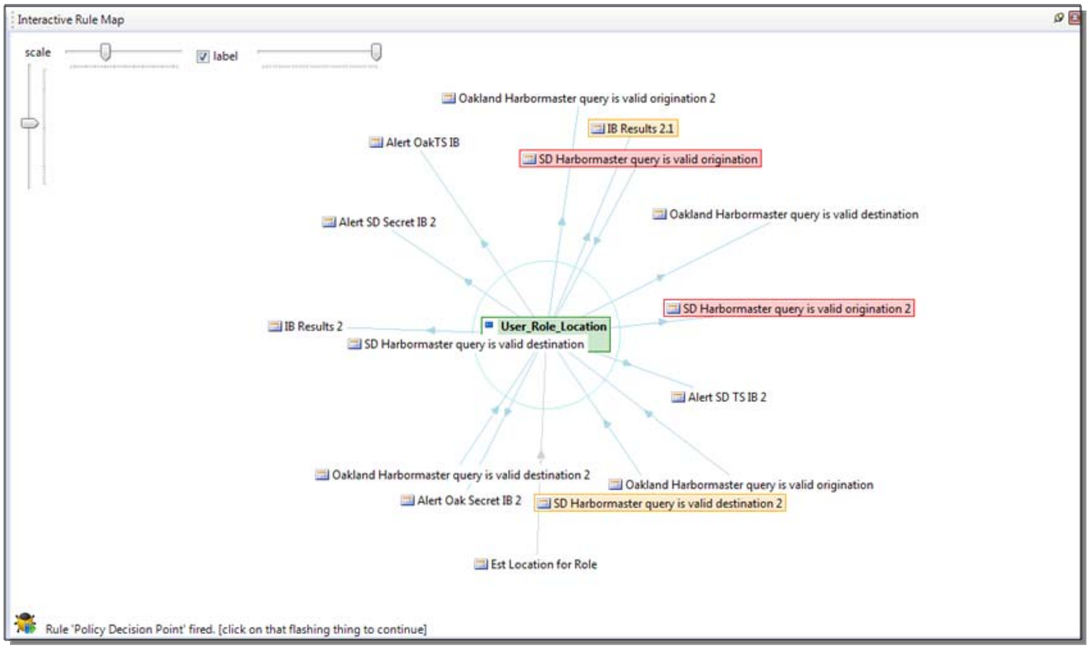
Rule 'Policy Decision Point' is in progress. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination 2
Policy Decision Point
PDP_Decision
True
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
SD Harbormaster query is valid destination 2
EWO query is valid 2
EWO query is valid
Oakland Harbormaster query is valid destination
Policy Decision Point 2

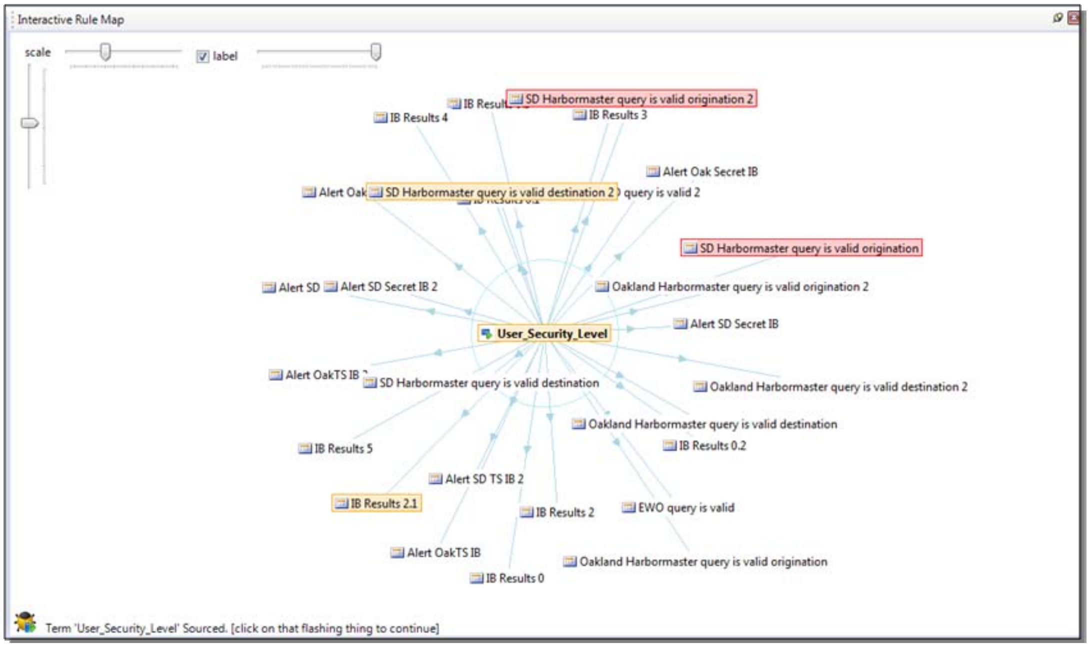
Rule 'Policy Decision Point' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Query is Valid
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid origination
HM_Valid_Query
True
SD Harbormaster query is valid origination
Query Invalid
Oakland Harbormaster quer
Oakland Harbormaster query is valid destination 2
HM query is invalid

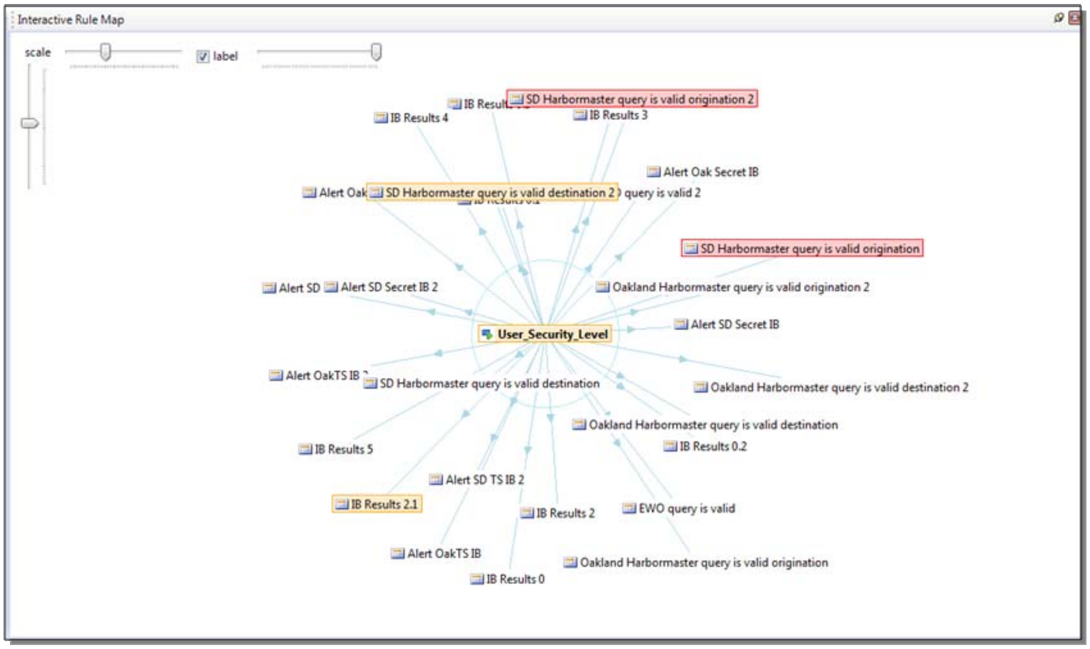
Rule 'Policy Decision Point' fired. [click on that flashing thing to continue]

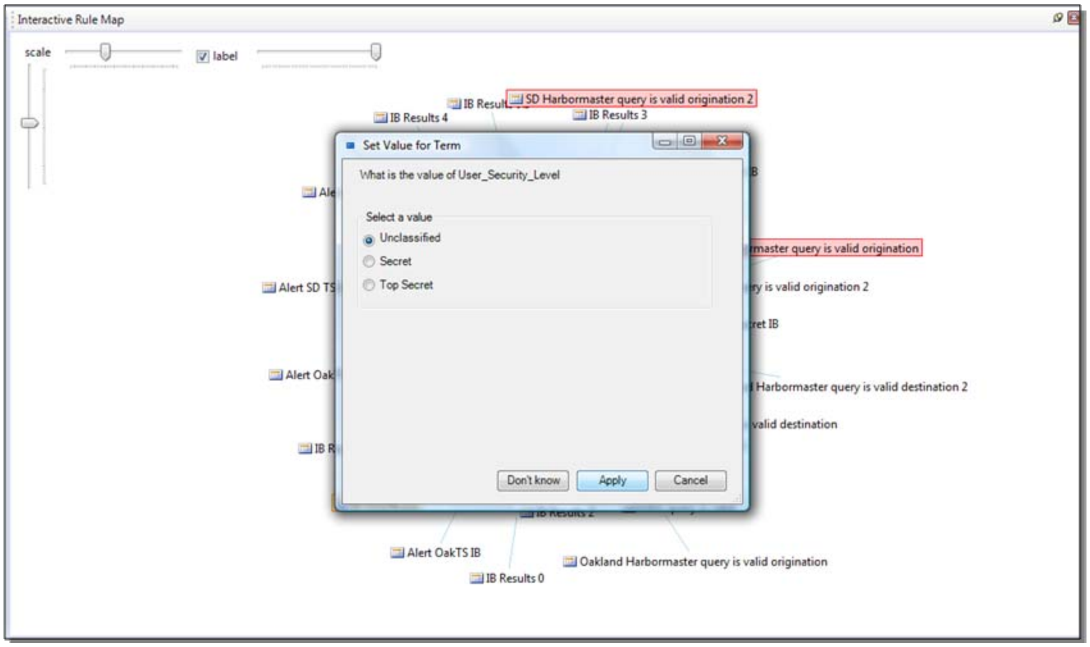

Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination 2
IB Results 2.1
Alert OakTS IB
SD Harbormaster query is valid origination
Alert SD Secret IB 2
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid origination 2
IB Results 2
User_Role_Location
SD Harbormaster query is valid destination
Alert SD TS IB 2
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination
Alert Oak Secret IB 2
SD Harbormaster query is valid destination 2
Est Location for Role

Rule 'Policy Decision Point' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
IB Resul
SD Harbormaster query is valid origination 2
IB Results 4
IB Results 3
Alert Oak Secret IB
Alert Oak
SD Harbormaster query is valid destination 2
query is valid 2
SD Harbormaster query is valid origination
Alert SD
Alert SD Secret IB 2
Oakland Harbormaster query is valid origination 2
Alert SD Secret IB
User_Security_Level
Alert OakTS IB
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
IB Results 5
IB Results 0.2
Alert SD TS IB 2
IB Results 2.1
IB Results 2
EWO query is valid
Alert OakTS IB
Oakland Harbormaster query is valid origination
IB Results 0

Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
IB Resul
SD Harbormaster query is valid origination 2
IB Results 4
IB Results 3
Alert Oak Secret IB
Alert Oak
SD Harbormaster query is valid destination 2
query is valid 2
SD Harbormaster query is valid origination
Alert SD
Alert SD Secret IB 2
Oakland Harbormaster query is valid origination 2
Alert SD Secret IB
User_Security_Level
Alert OakTS IB
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
IB Results 5
IB Results 0.2
Alert SD TS IB 2
IB Results 2.1
IB Results 2
EWO query is valid
Alert OakTS IB
Oakland Harbormaster query is valid origination
IB Results 0


Interactive Rule Map
scale
☑ label
IB Results 4
IB Resul
SD Harbormaster query is valid origination 2
IB Results 3
Ale
B
rmaster query is valid origination
Alert SD TS
ry is valid origination 2
ret IB
Alert Oak
Harbormaster query is valid destination 2
valid destination
IB R
IB Results 2
Alert OakTS IB
IB Results 0
Oakland Harbormaster query is valid origination

Set Value for Term

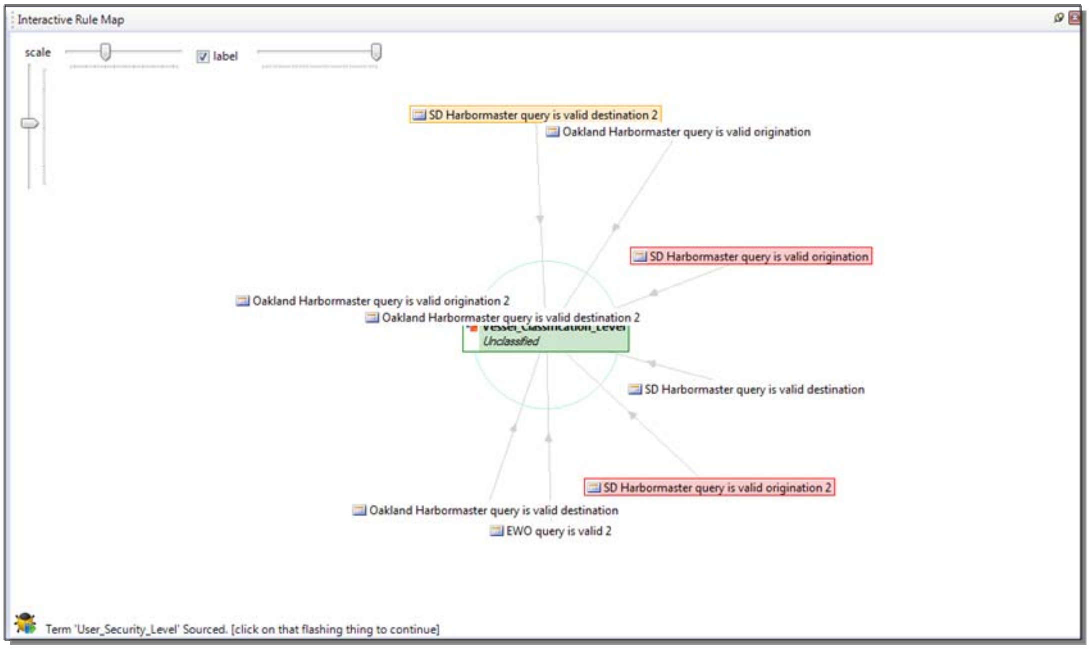
What is the value of User_Security_Level

Select a value

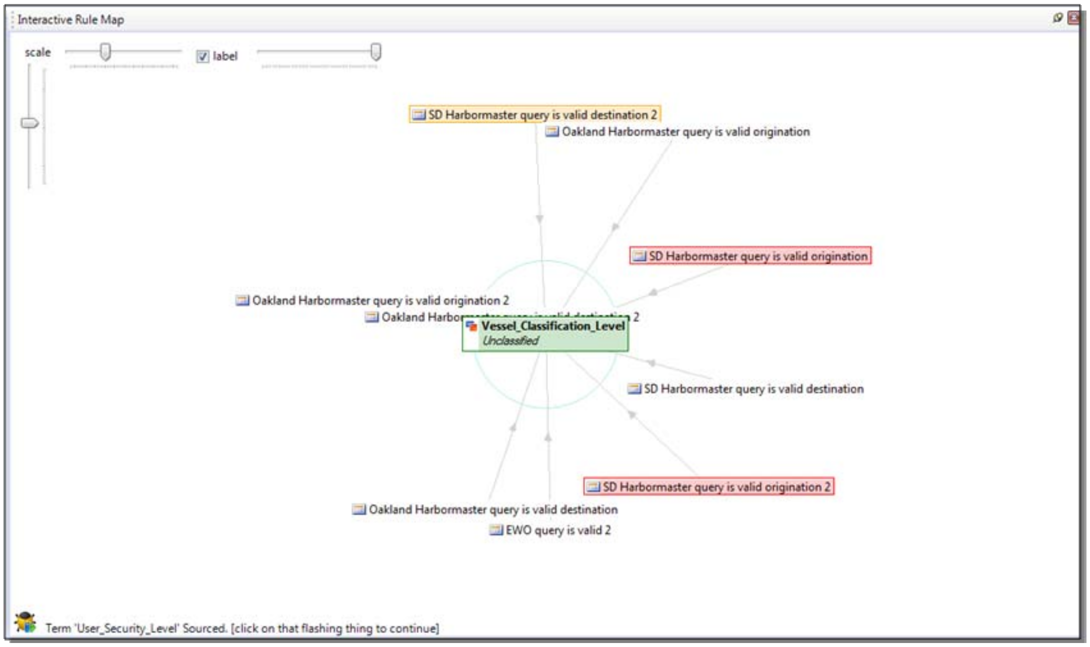
● Unclassified

○ Secret

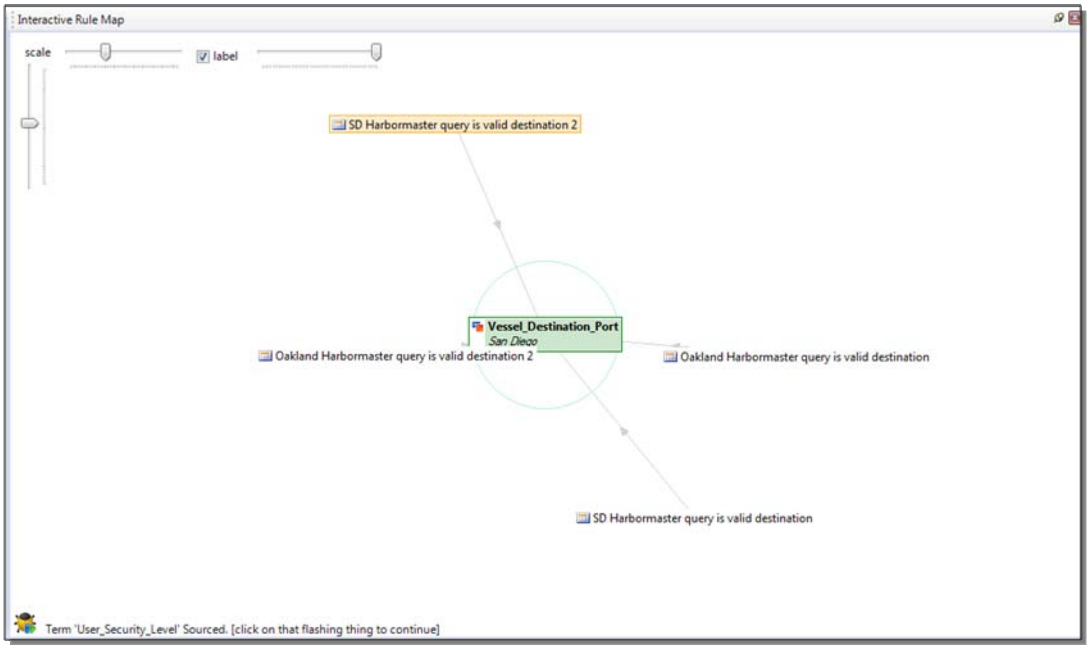
○ Top Secret


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination 2
Vessel_Classification_Level
Unclassified
SD Harbormaster query is valid destination
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination
EWO query is valid 2

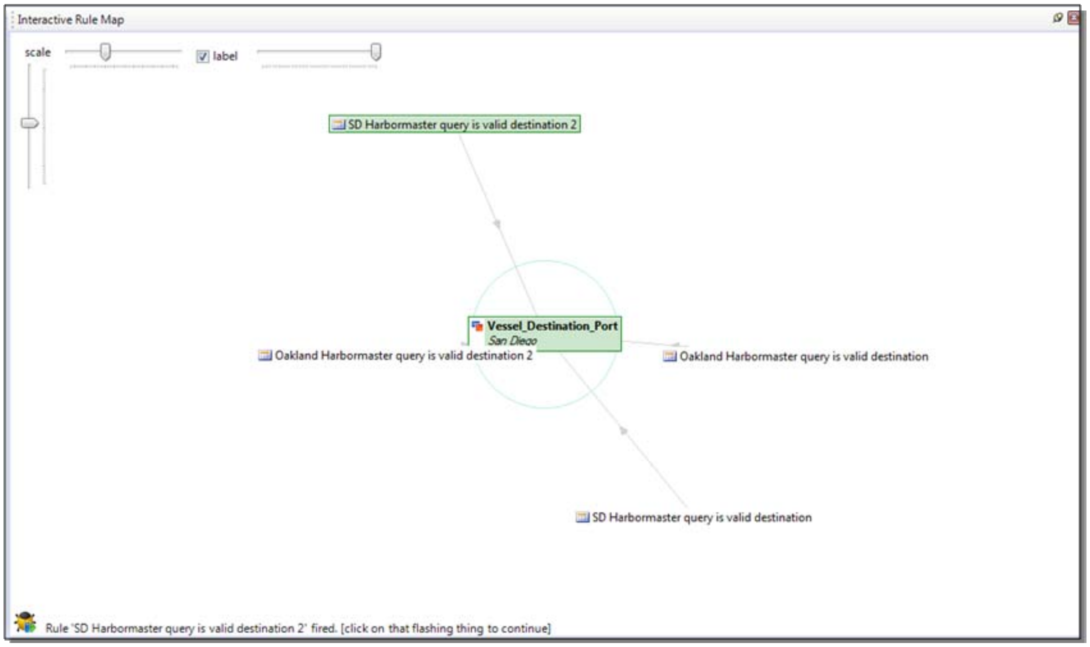
Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination 2
Vessel_Classification_Level
Unclassified
SD Harbormaster query is valid destination
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination
EWO query is valid 2

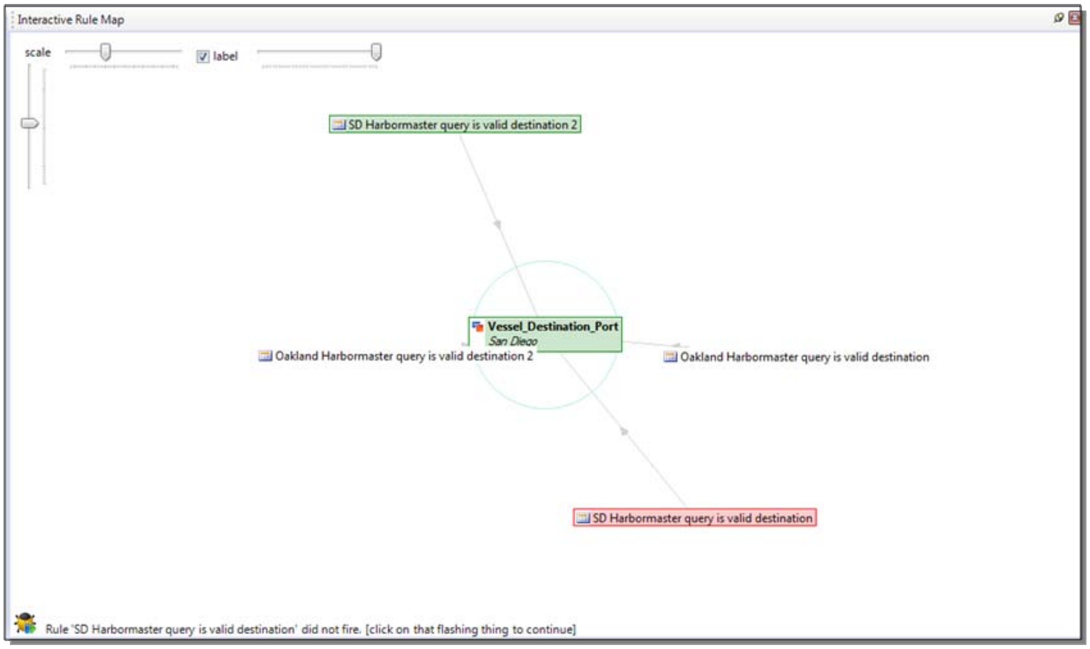
Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination

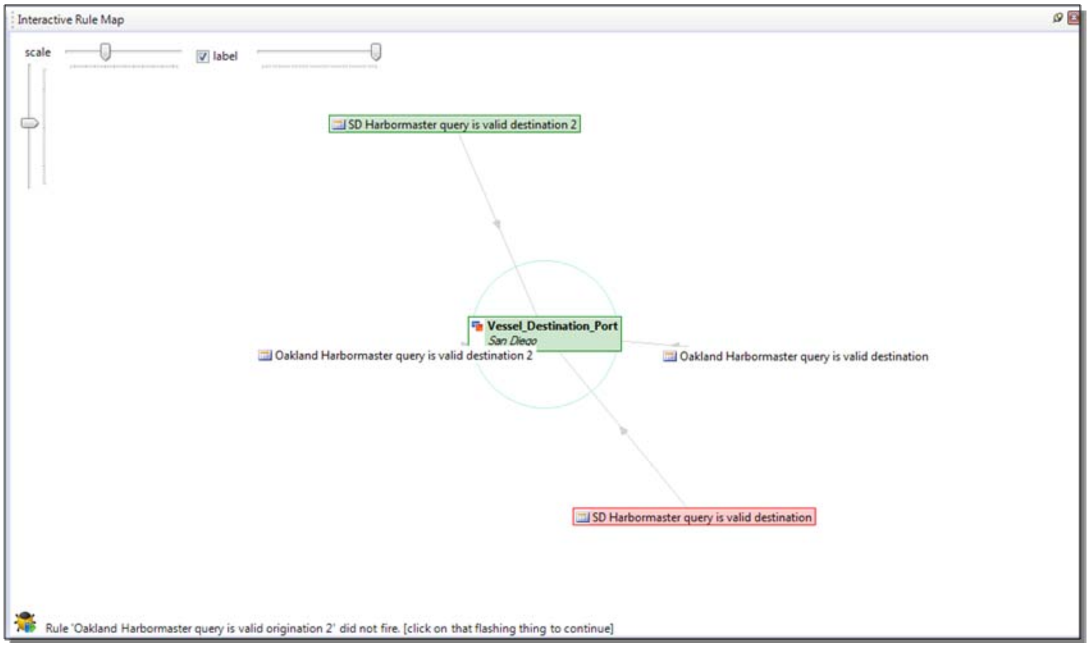
Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination

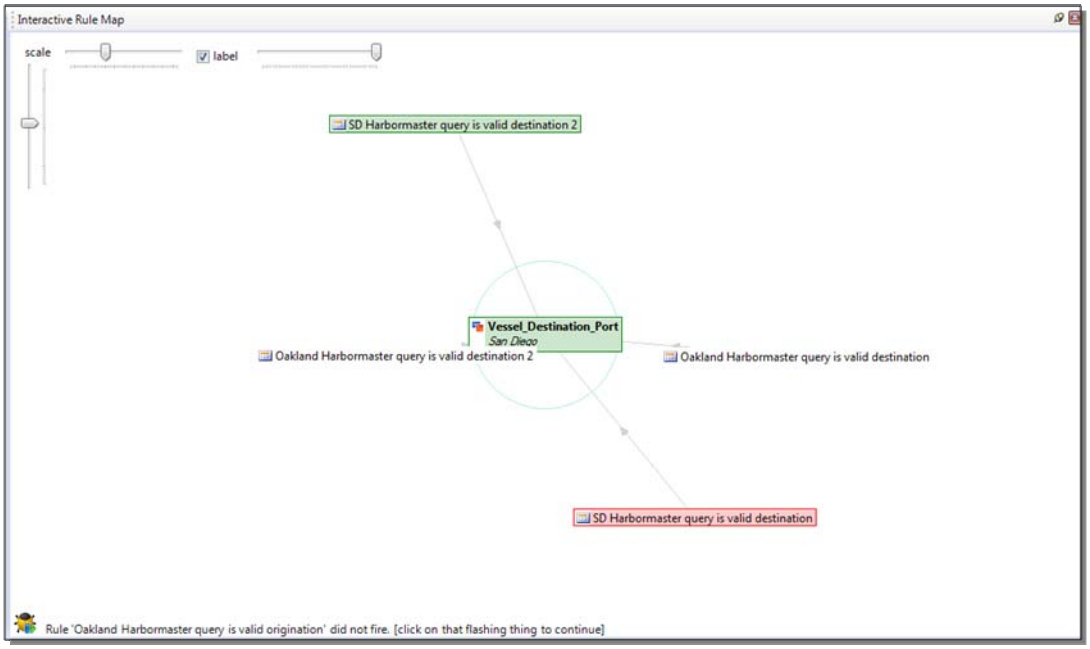
Rule 'SD Harbormaster query is valid destination 2' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination

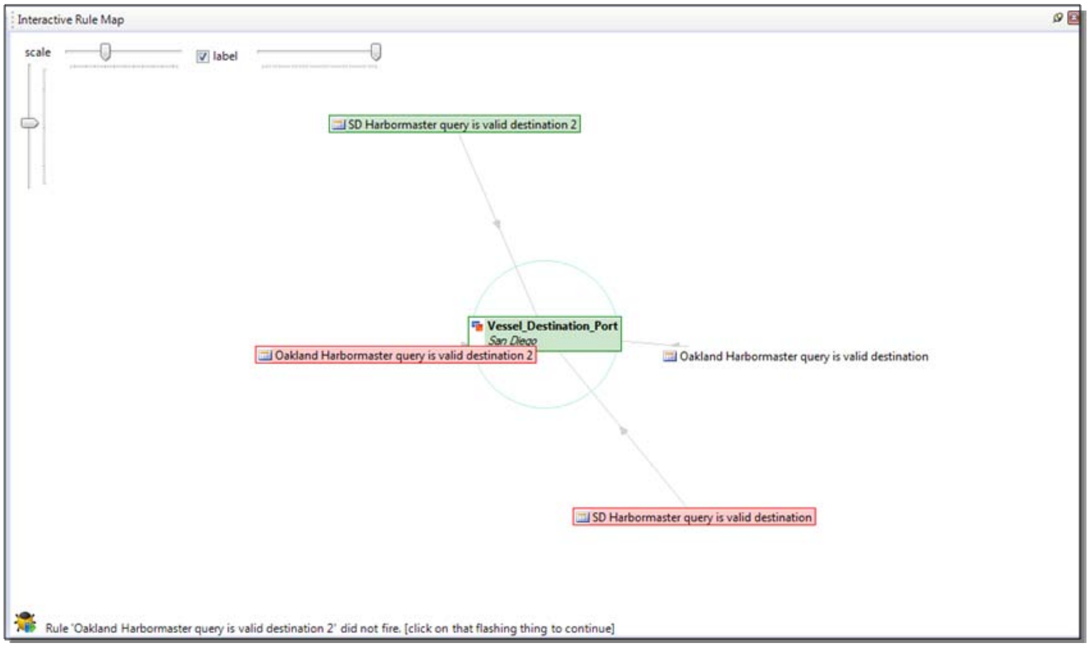
Rule 'SD Harbormaster query is valid destination' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination

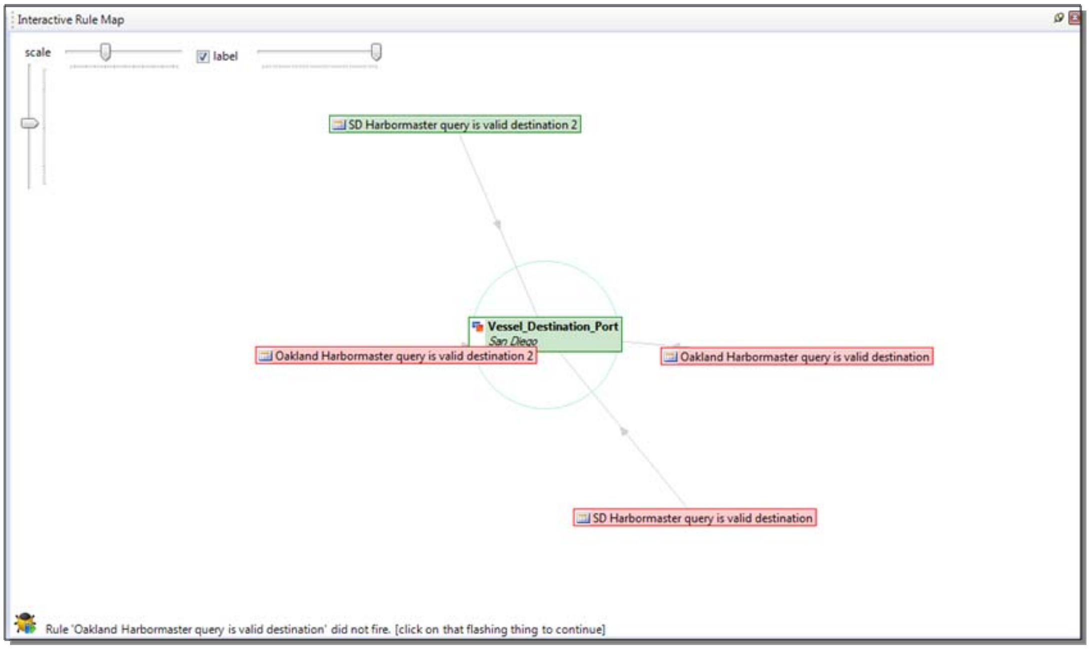
Rule 'Oakland Harbormaster query is valid origination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination

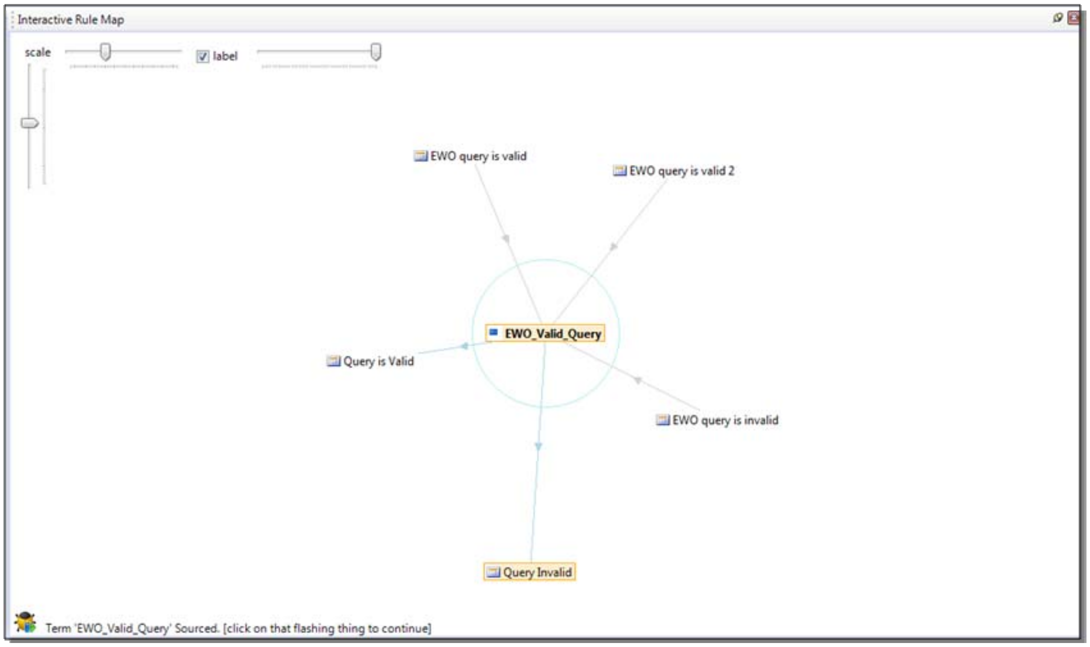
Rule 'Oakland Harbormaster query is valid origination' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination

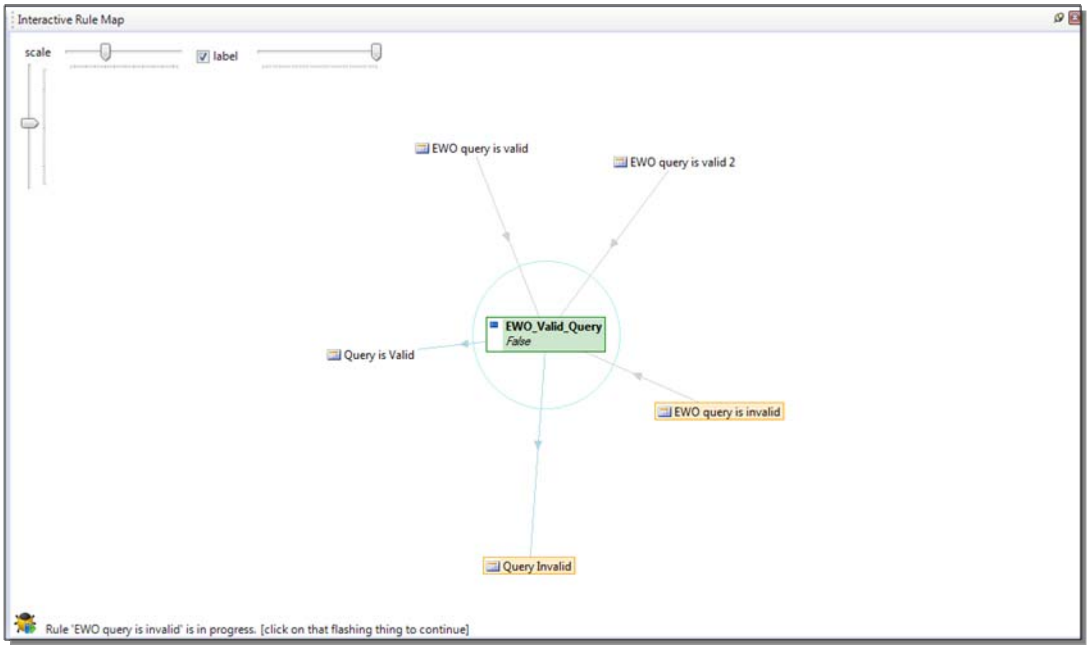
Rule 'Oakland Harbormaster query is valid destination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination

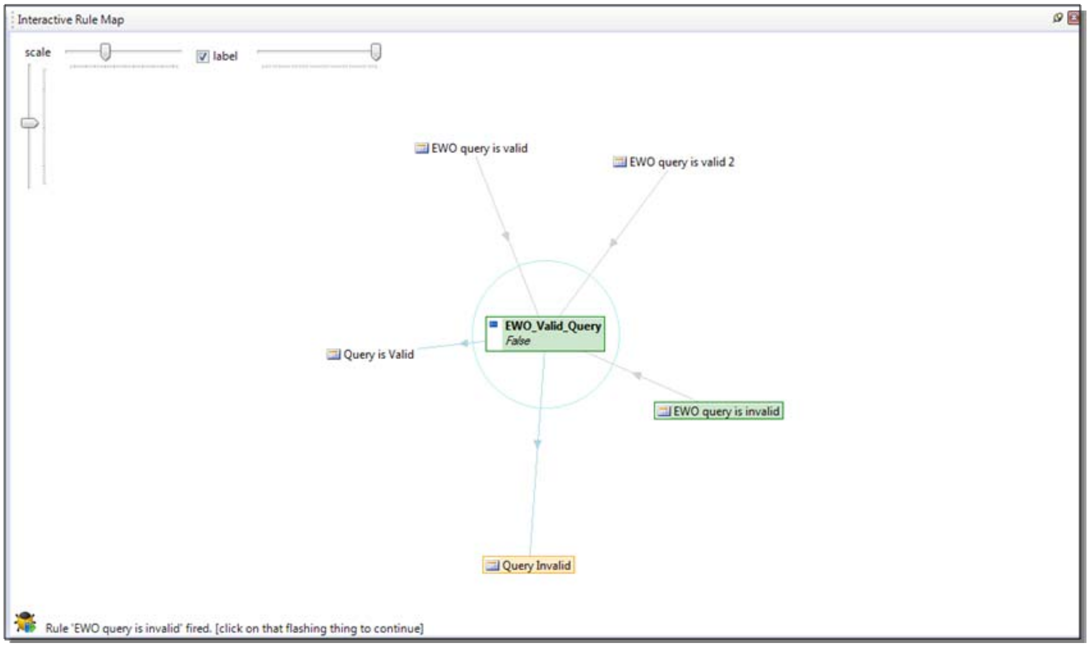
Rule 'Oakland Harbormaster query is valid destination' did not fire. [click on that flashing thing to continue]



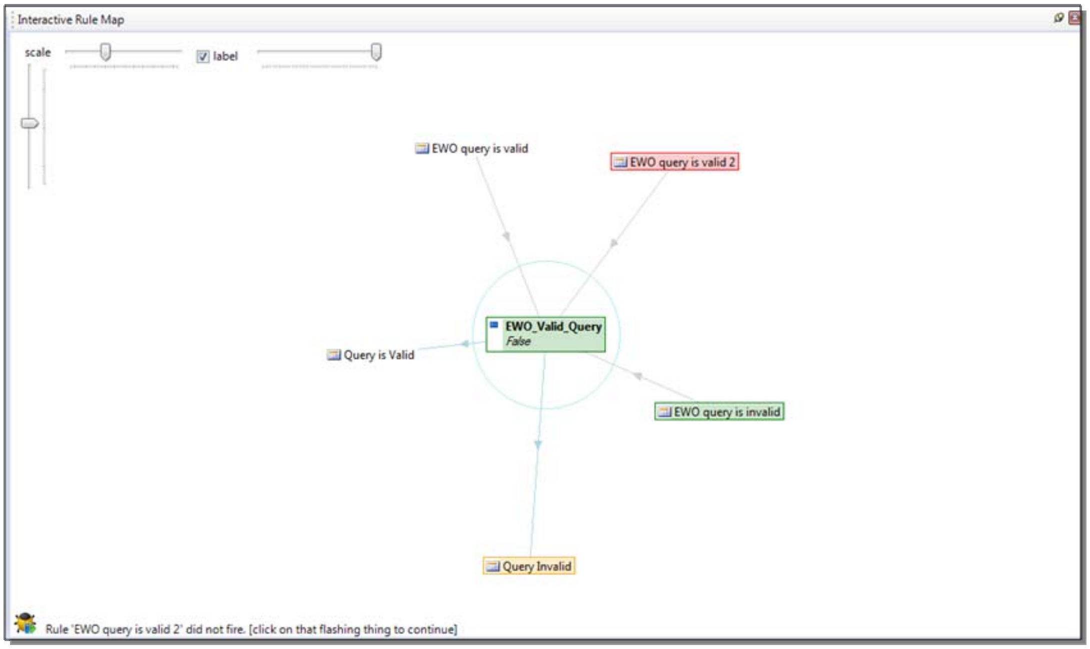
Term 'EWO_Valid_Query' Sourced. [click on that flashing thing to continue]



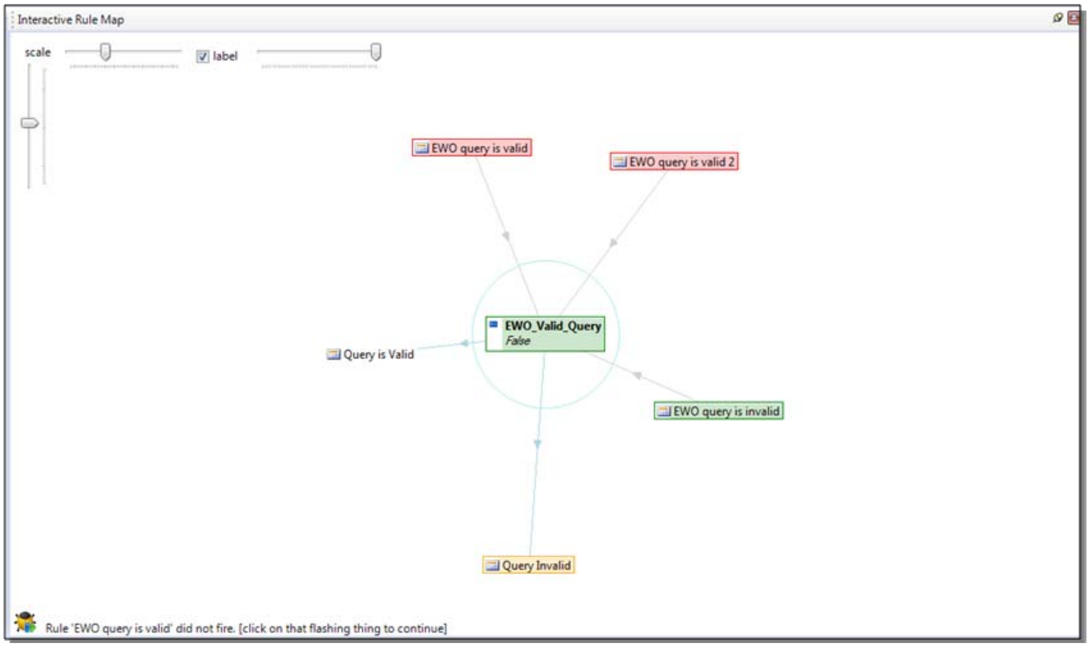
Rule 'EWO query is invalid' is in progress. [click on that flashing thing to continue]



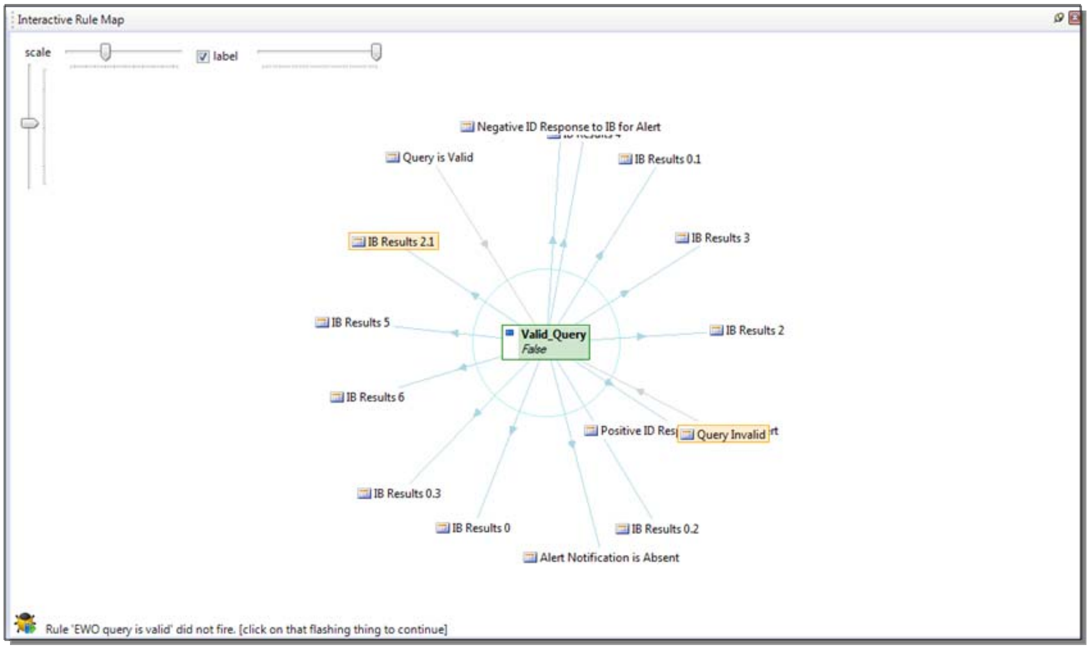
Rule 'EWO query is invalid' fired. [click on that flashing thing to continue]



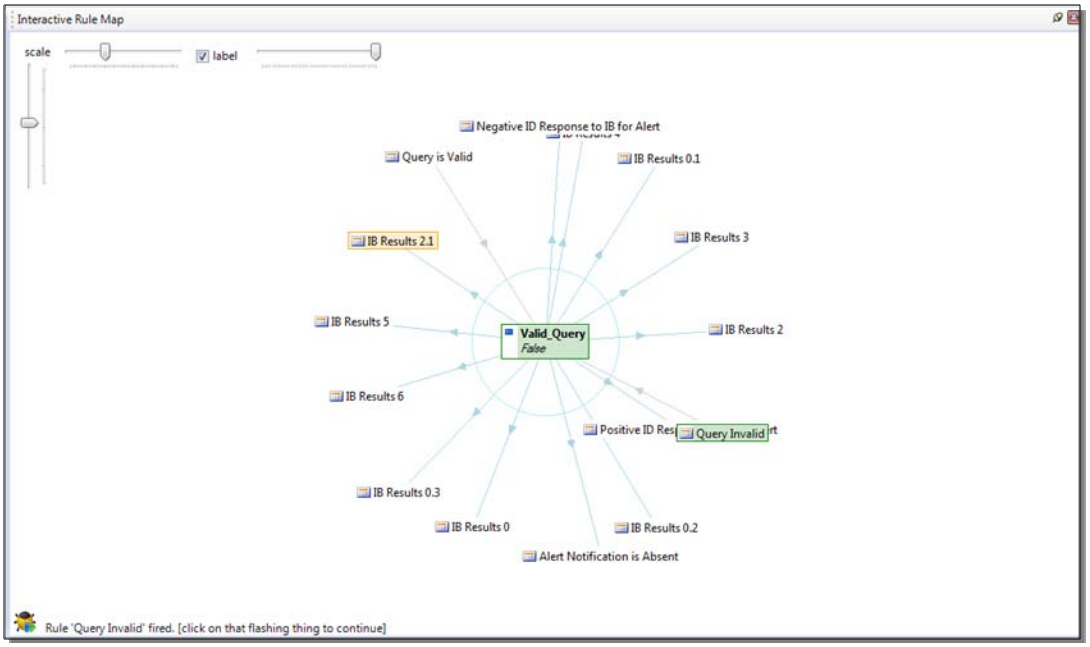
Rule 'EWO query is valid 2' did not fire. [click on that flashing thing to continue]



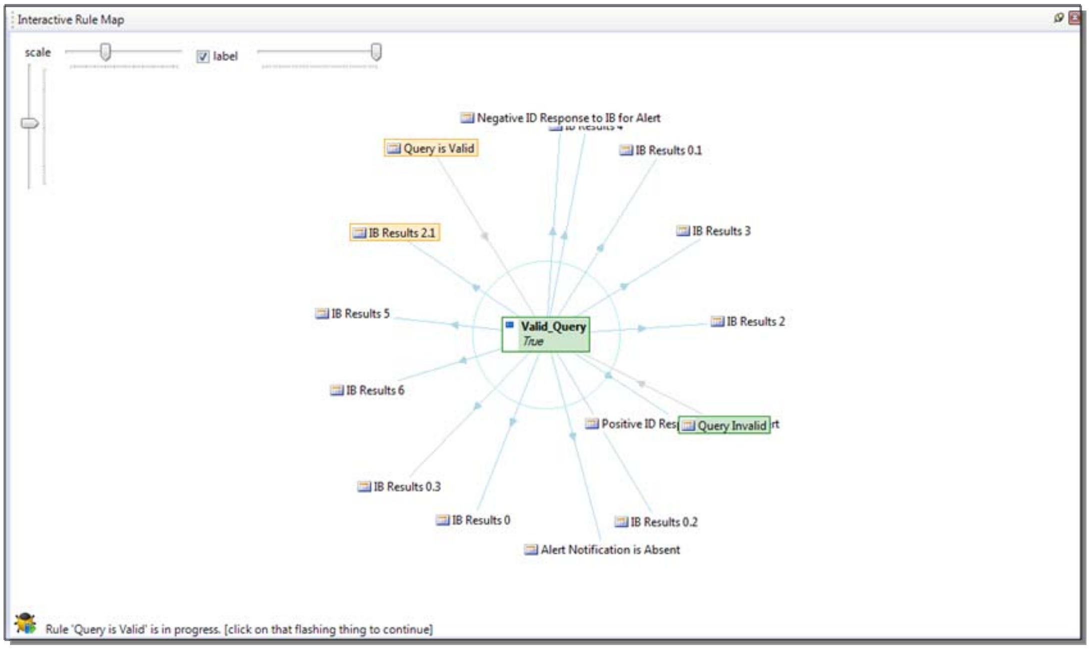
Rule 'EWO query is valid' did not fire. [click on that flashing thing to continue]



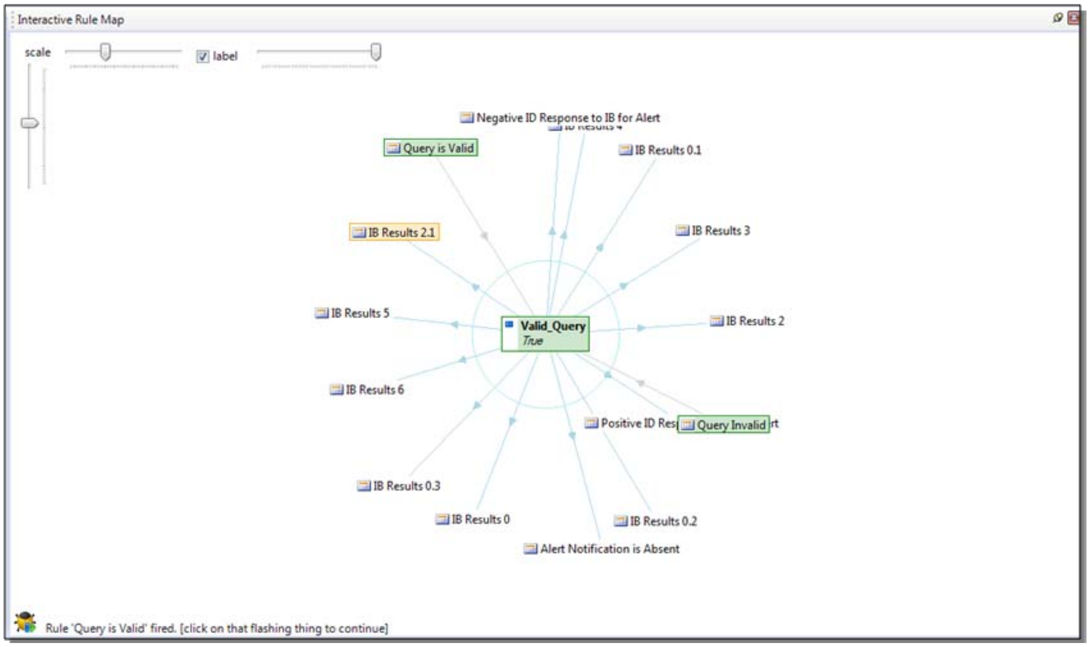
Rule 'EWO query is valid' did not fire. [click on that flashing thing to continue]



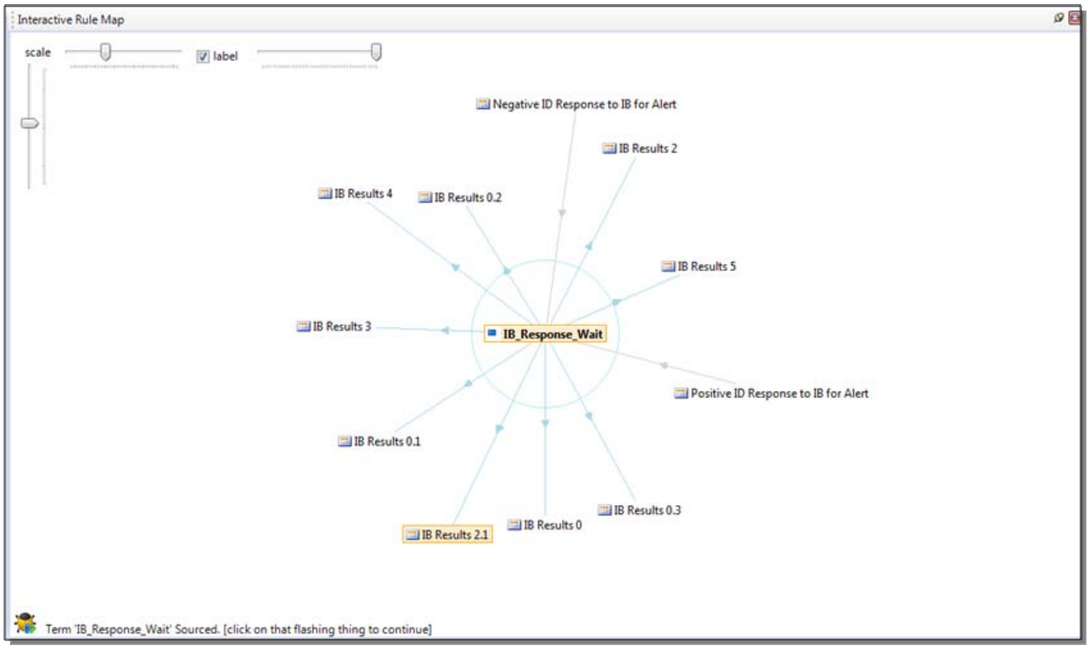
Rule 'Query Invalid' fired. [click on that flashing thing to continue]



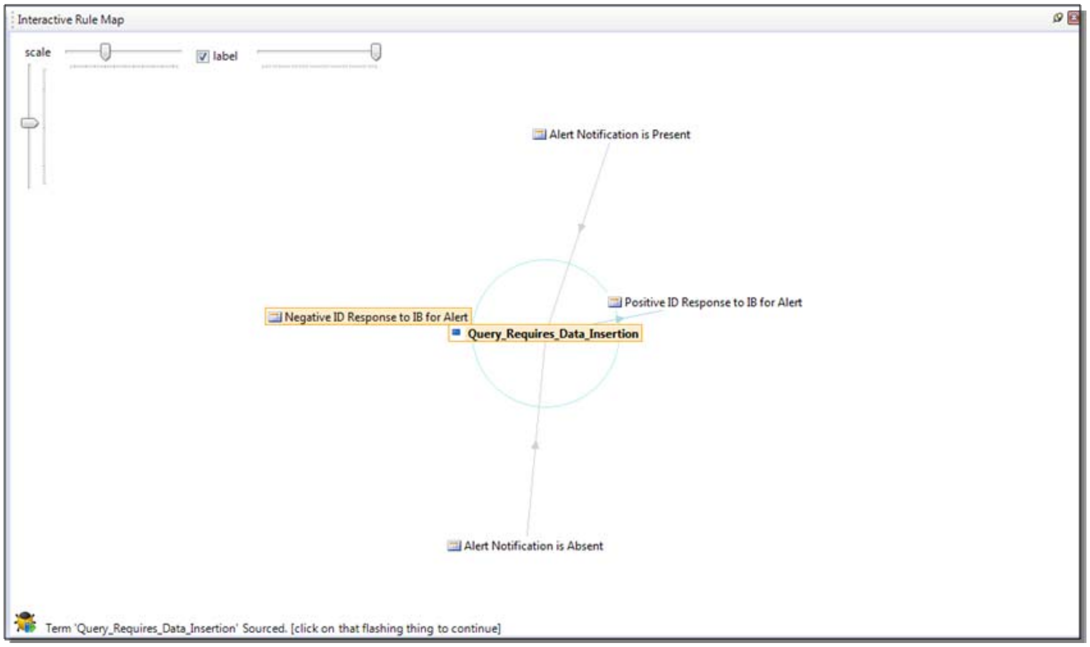
Rule 'Query is Valid' is in progress. [click on that flashing thing to continue]



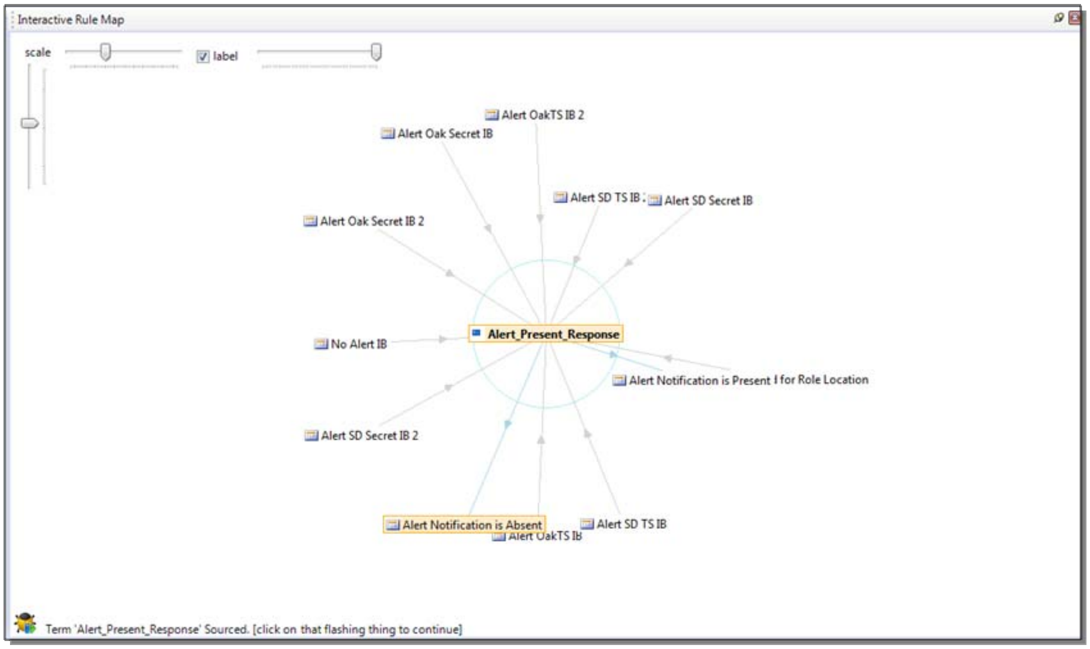
Rule 'Query is Valid' fired. [click on that flashing thing to continue]



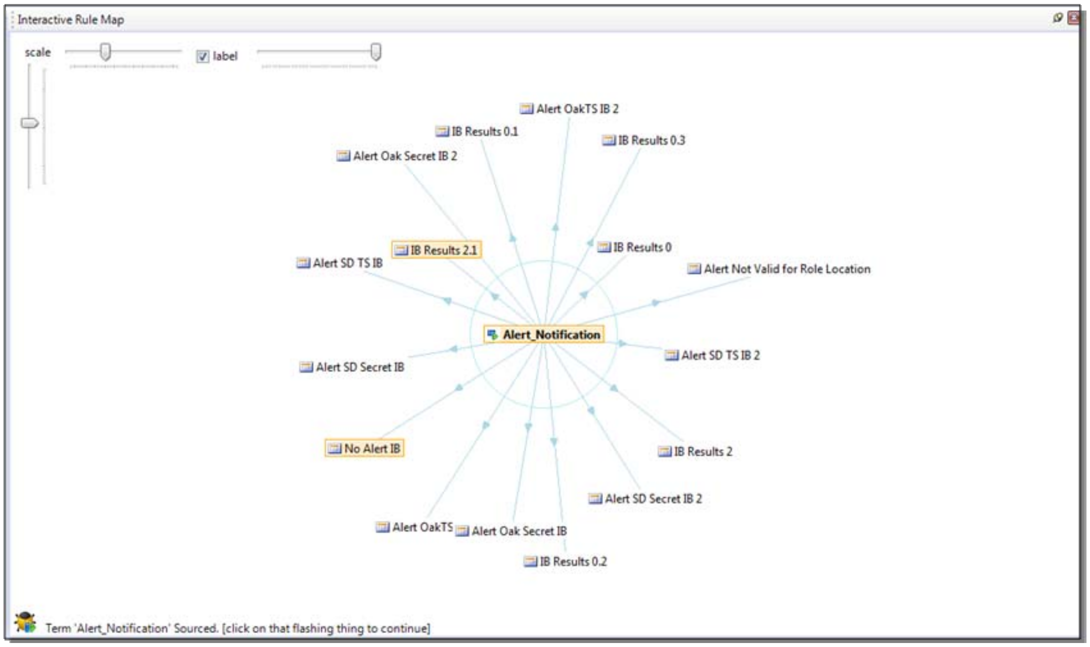
Term 'IB_Response_Wait' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Alert Notification is Present
Positive ID Response to IB for Alert
Negative ID Response to IB for Alert
Query_Requires_Data_Insertion
Alert Notification is Absent

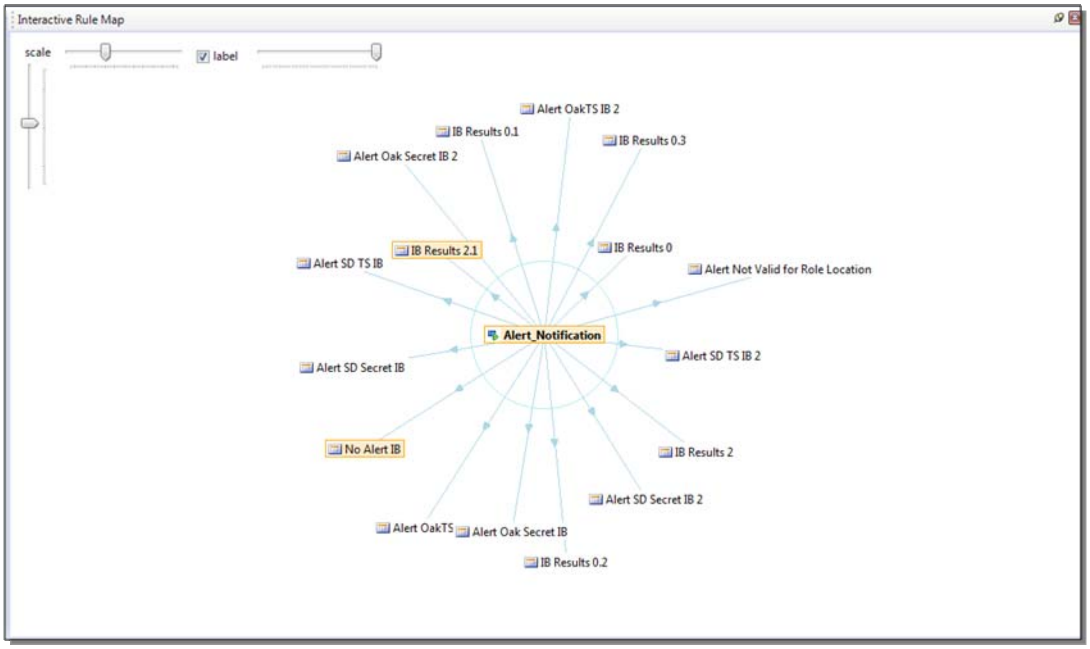
Term 'Query_Requires_Data_Insertion' Sourced. [click on that flashing thing to continue]

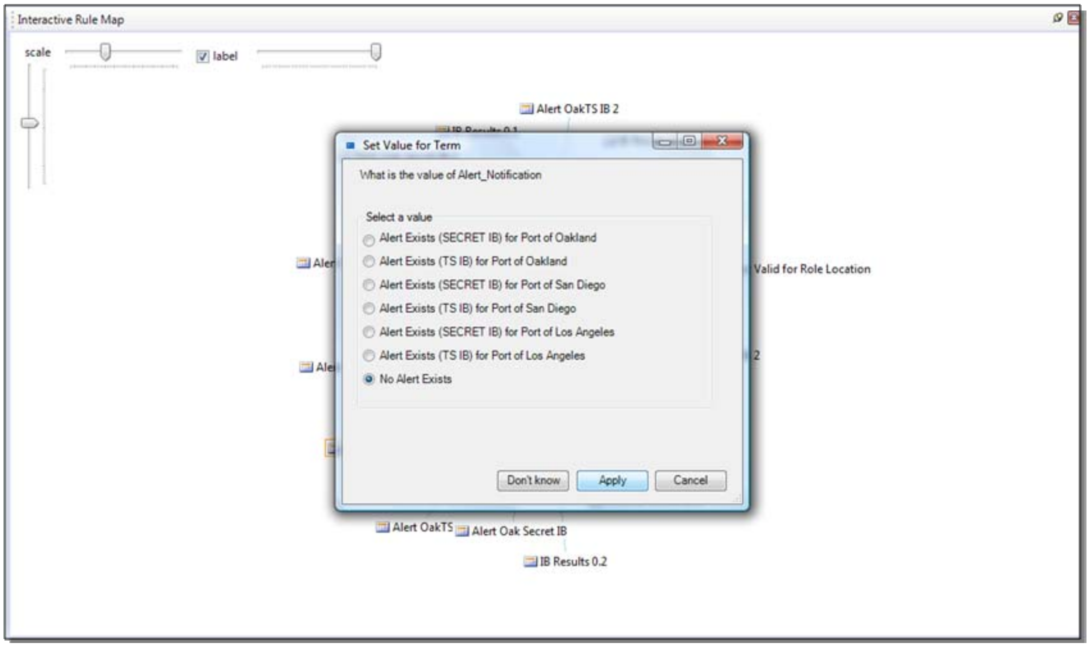


Term 'Alert_Present_Response' Sourced. [click on that flashing thing to continue]



Term 'Alert_Notification' Sourced. [click on that flashing thing to continue]



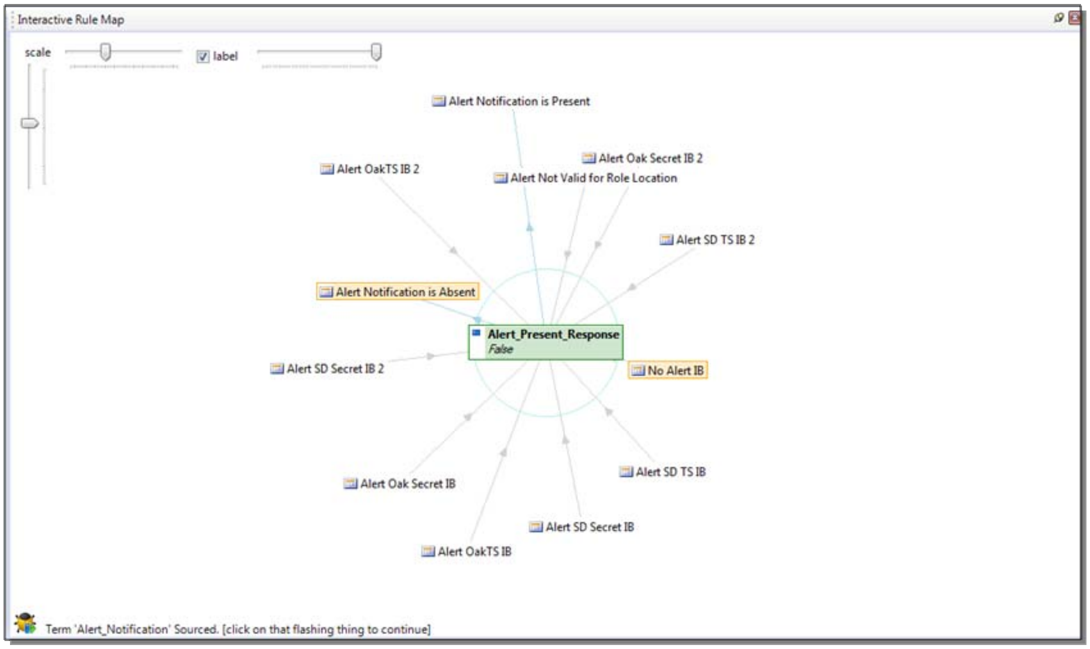


Set Value for Term

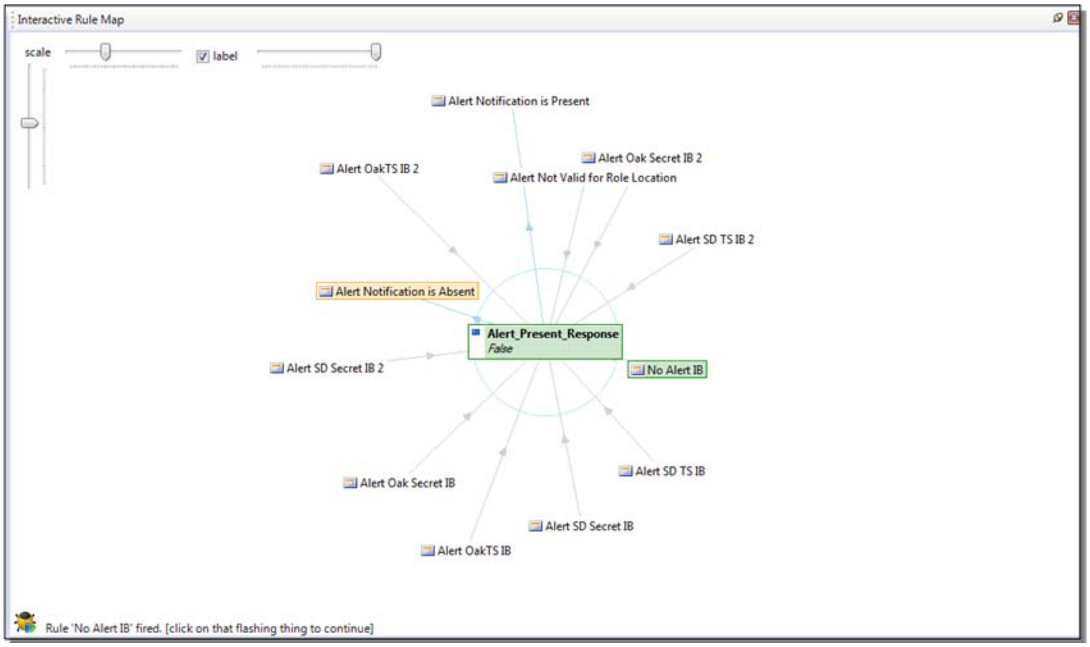
What is the value of Alert_Notification

Select a value

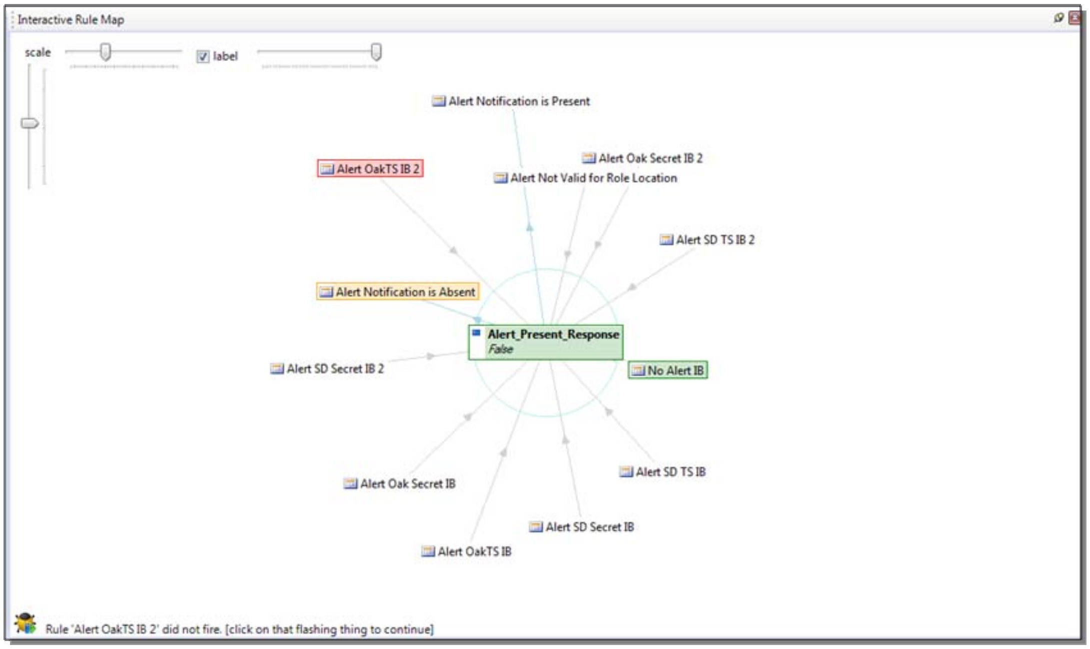
○ Alert Exists (SECRET IB) for Port of Oakland
○ Alert Exists (TS IB) for Port of Oakland
○ Alert Exists (SECRET IB) for Port of San Diego
○ Alert Exists (TS IB) for Port of San Diego
○ Alert Exists (SECRET IB) for Port of Los Angeles
○ Alert Exists (TS IB) for Port of Los Angeles
● No Alert Exists



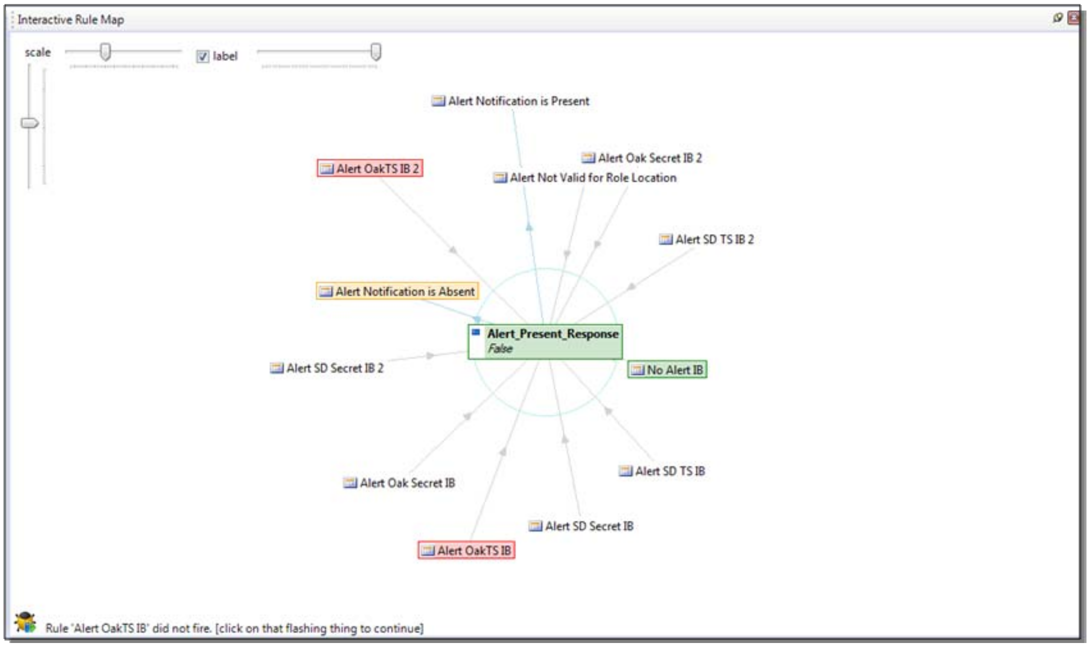
Term 'Alert_Notification' Sourced. [click on that flashing thing to continue]



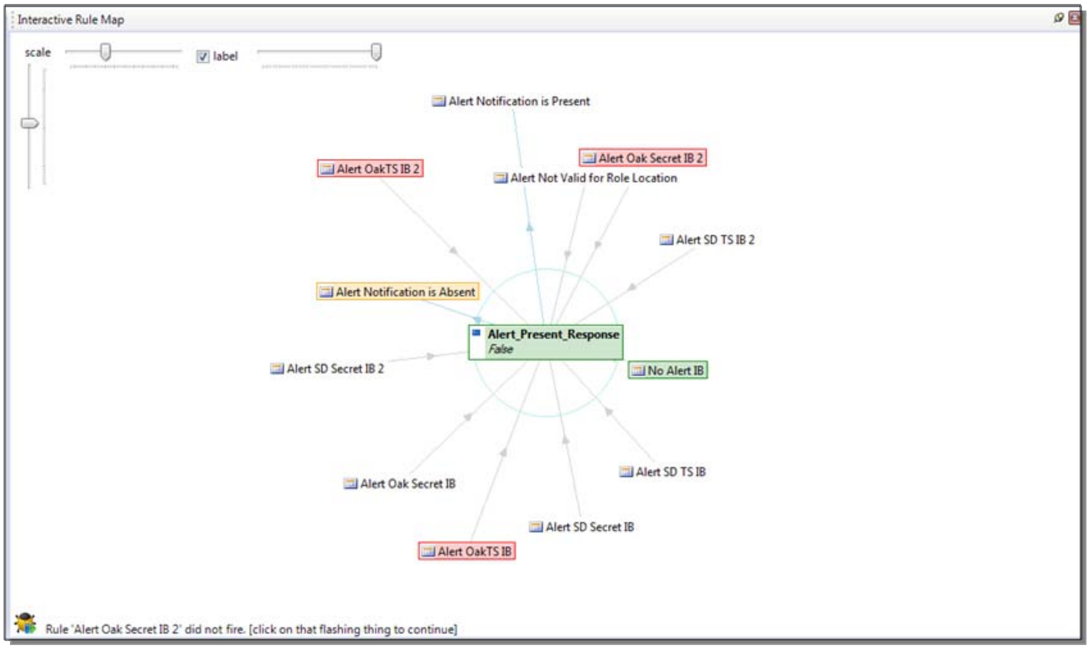
Rule 'No Alert IB' fired. [click on that flashing thing to continue]



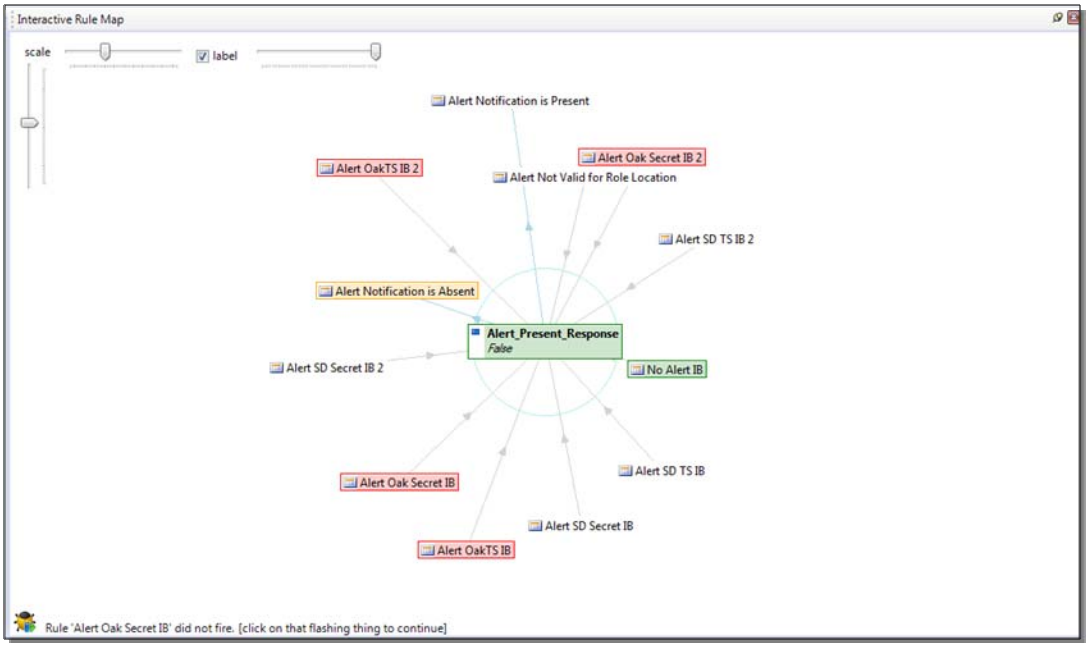
Rule 'Alert OakTS IB 2' did not fire. [click on that flashing thing to continue]



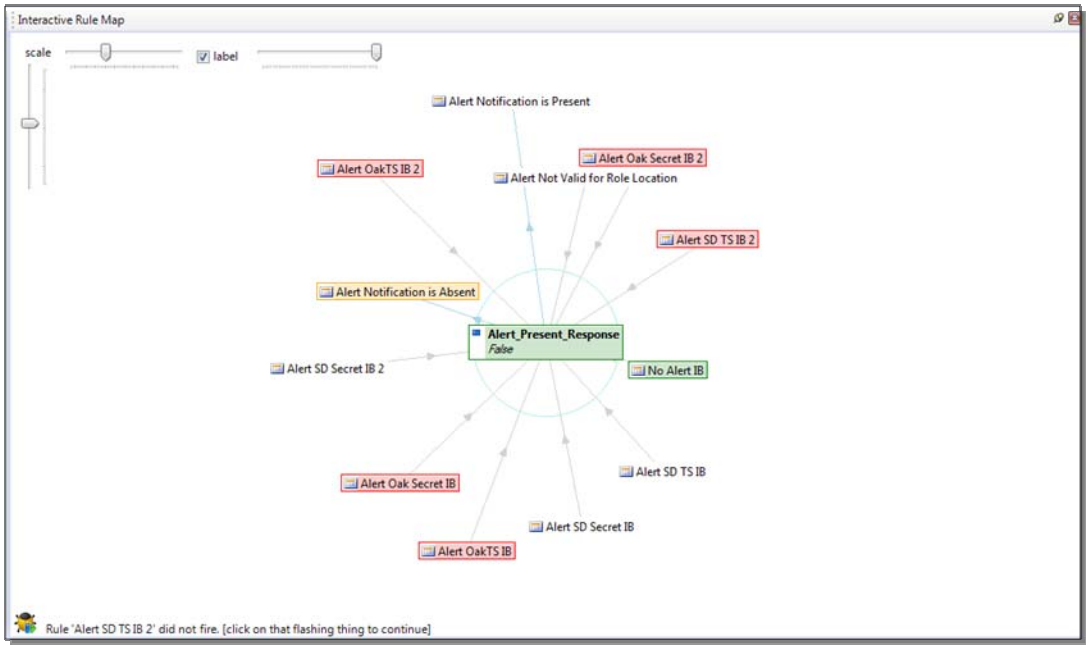
Rule 'Alert OakTS IB' did not fire. [click on that flashing thing to continue]



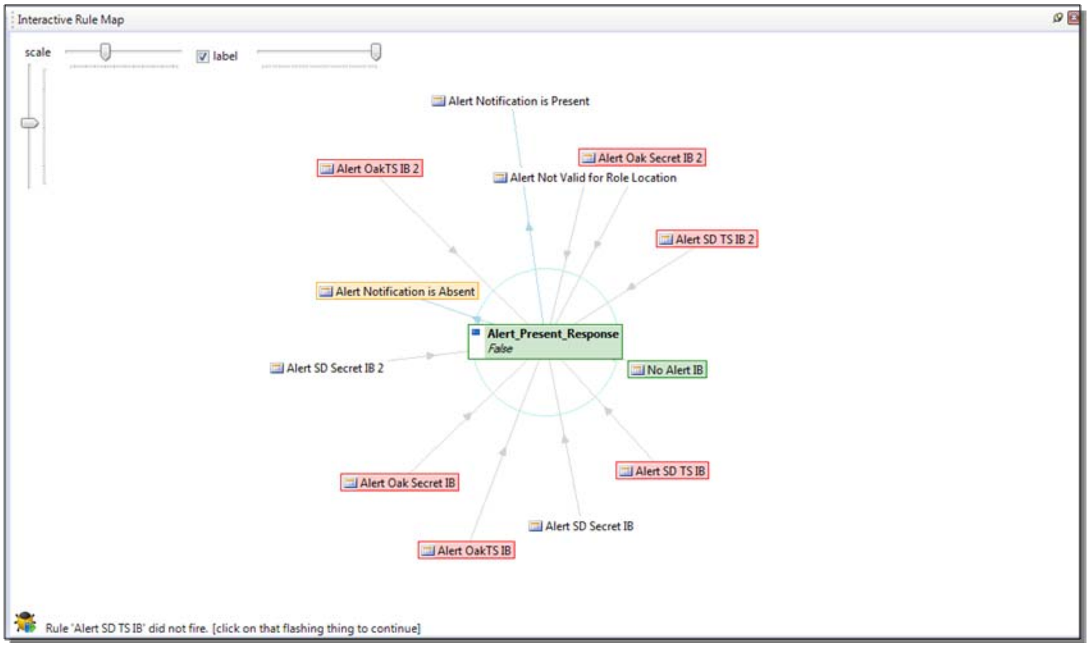
Rule 'Alert Oak Secret IB 2' did not fire. [click on that flashing thing to continue]



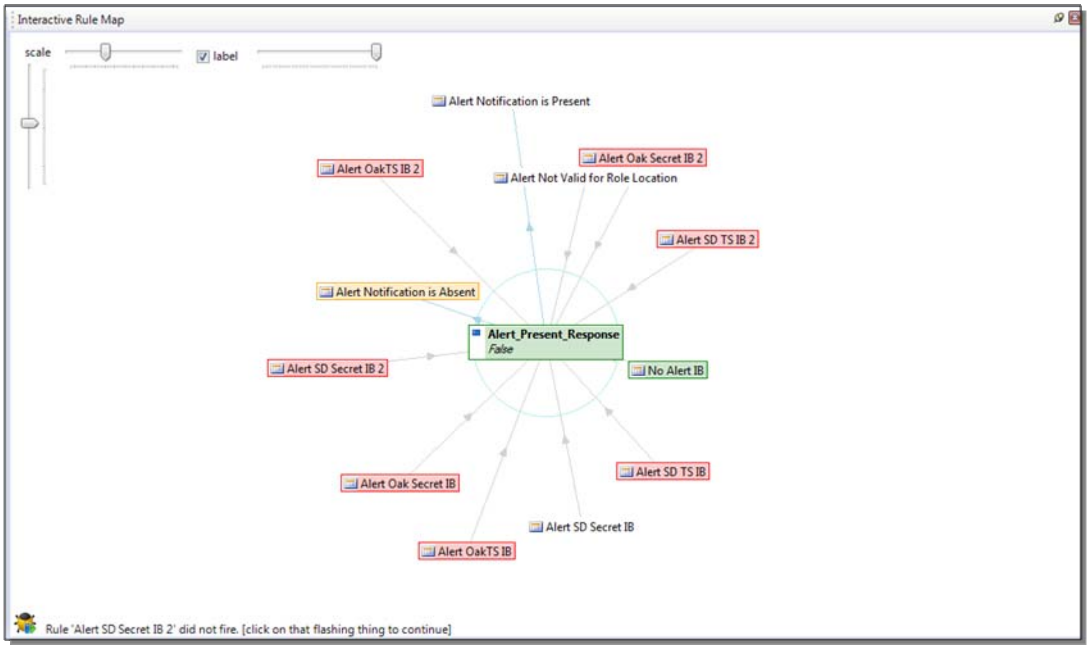
Rule 'Alert Oak Secret IB' did not fire. [click on that flashing thing to continue]



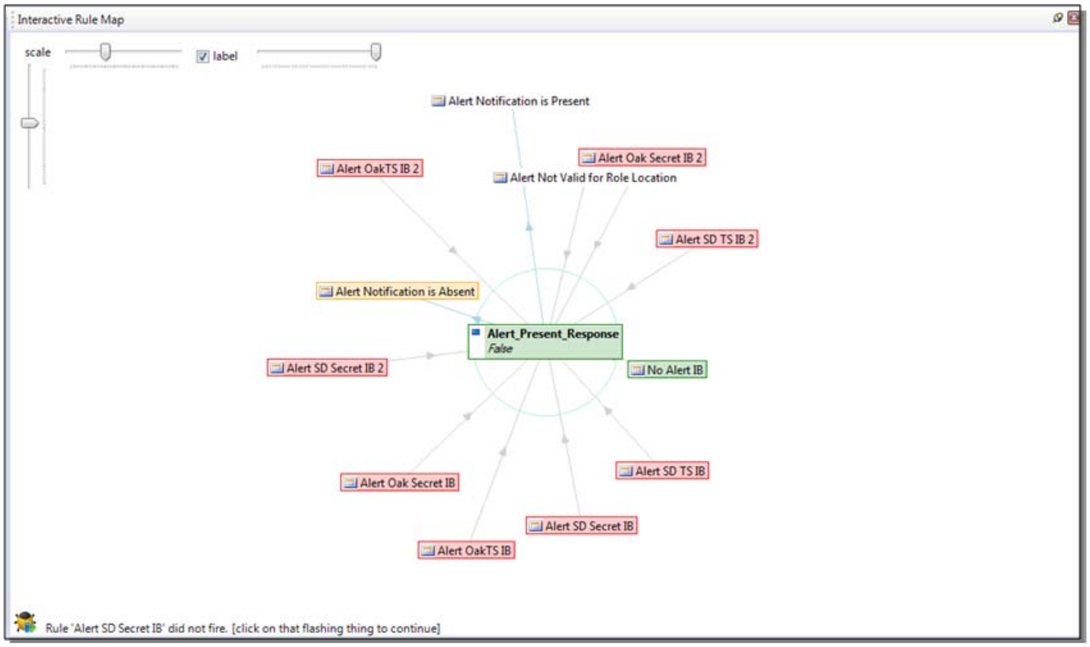
Rule 'Alert SD TS IB 2' did not fire. [click on that flashing thing to continue]



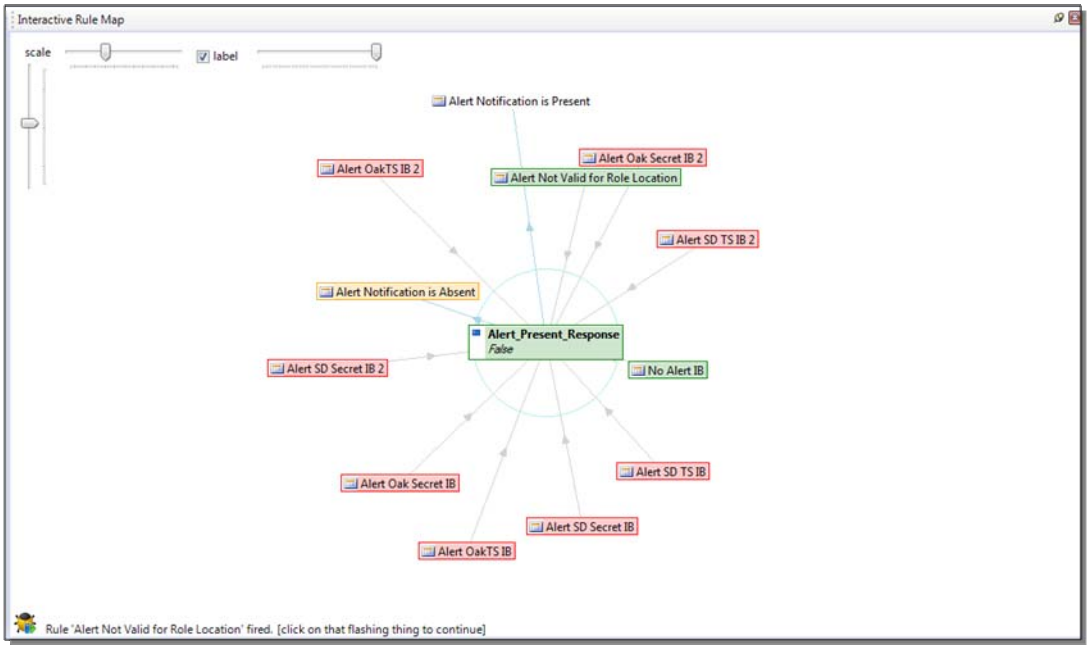
Rule 'Alert SD TS IB' did not fire. [click on that flashing thing to continue]



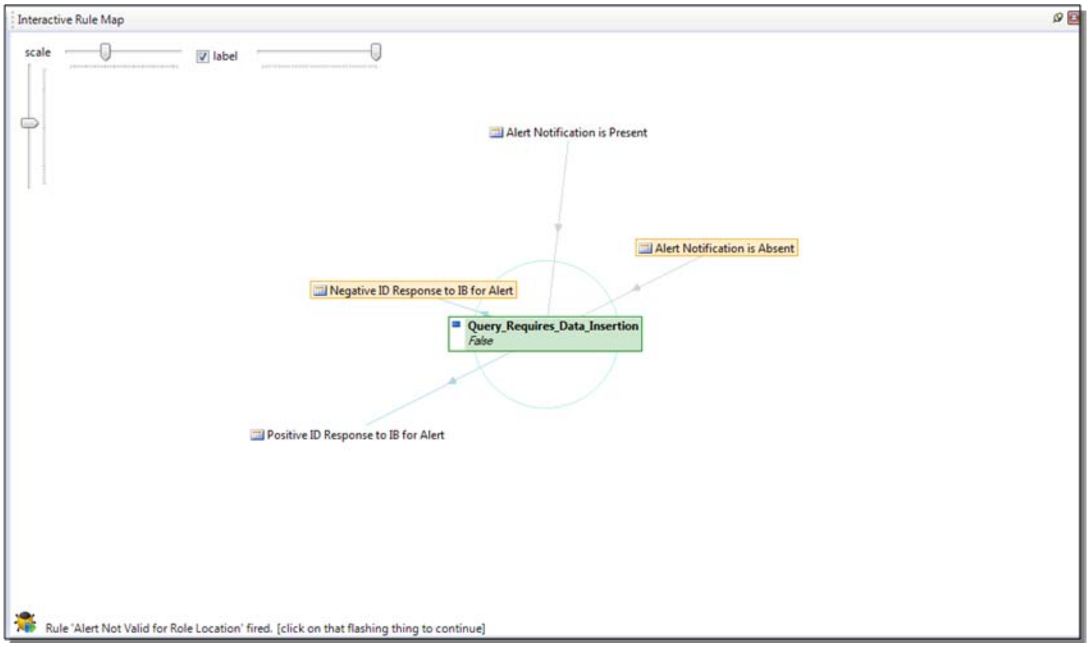
Rule 'Alert SD Secret IB 2' did not fire. [click on that flashing thing to continue]



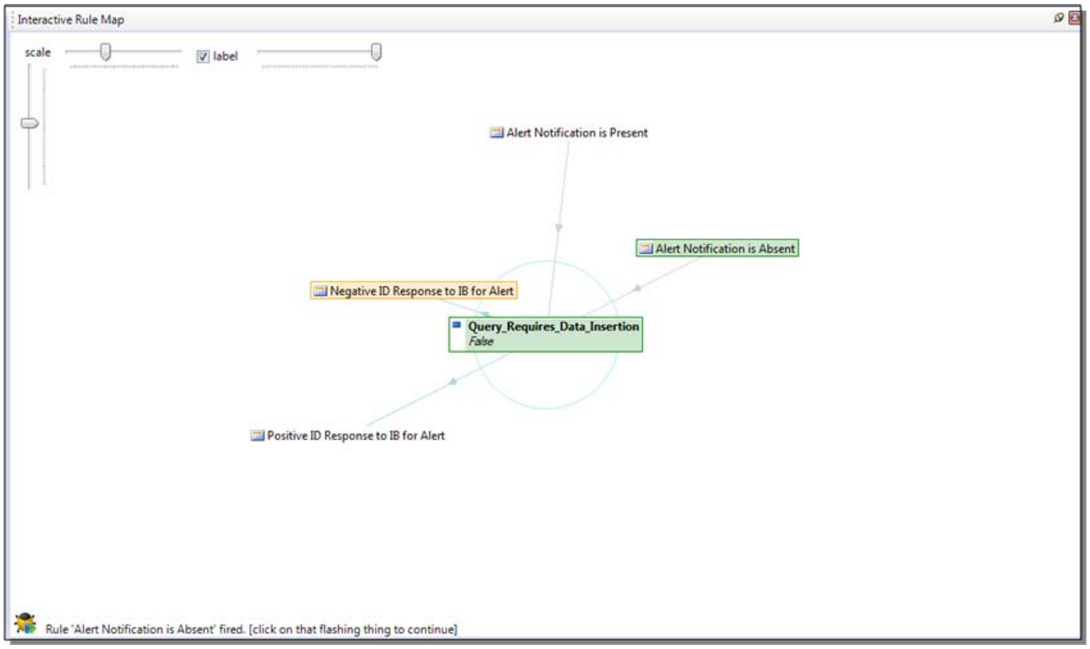
Rule 'Alert SD Secret IB' did not fire. [click on that flashing thing to continue]



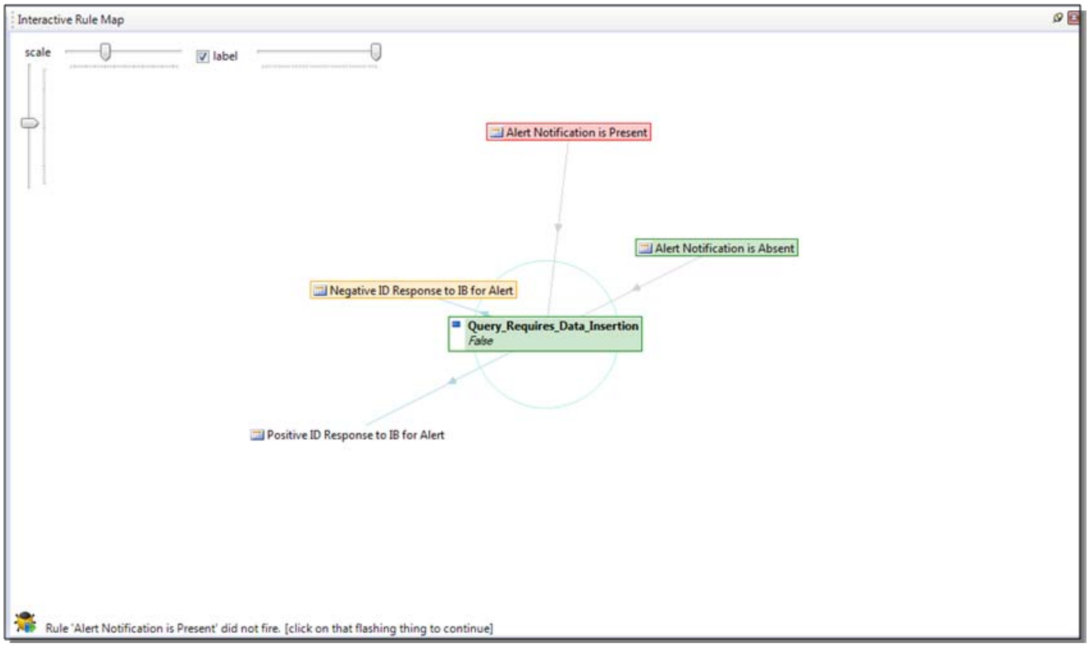
Rule 'Alert Not Valid for Role Location' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Alert Notification is Present
Alert Notification is Absent
Negative ID Response to IB for Alert
Query_Requires_Data_Insertion
False
Positive ID Response to IB for Alert

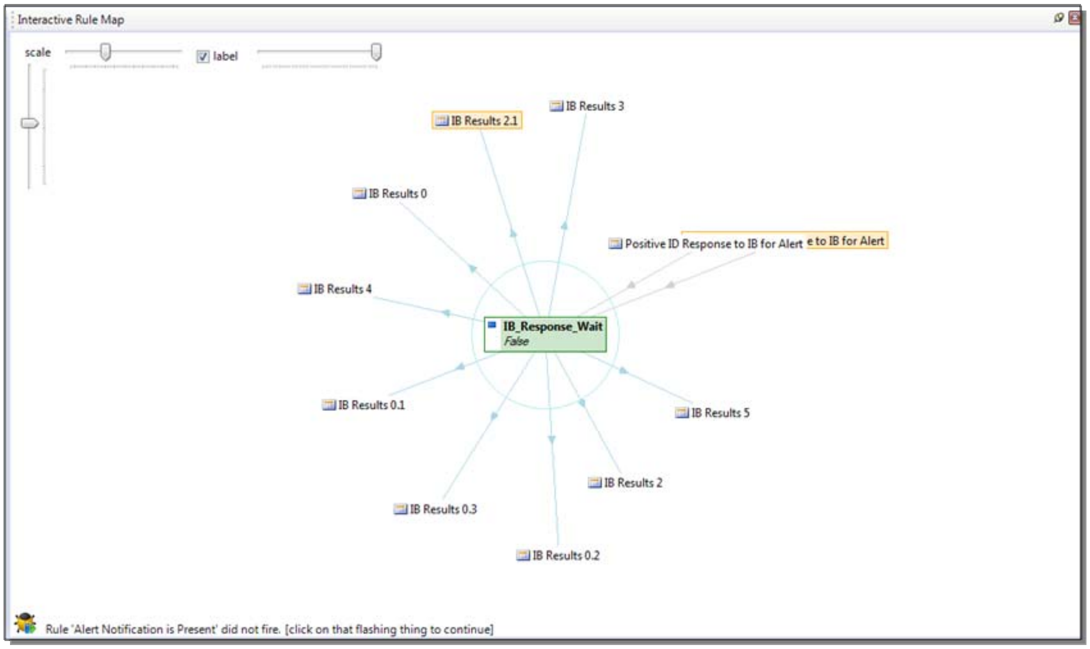
Rule 'Alert Not Valid for Role Location' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Alert Notification is Present
Alert Notification is Absent
Negative ID Response to IB for Alert
Query_Requires_Data_Insertion
False
Positive ID Response to IB for Alert

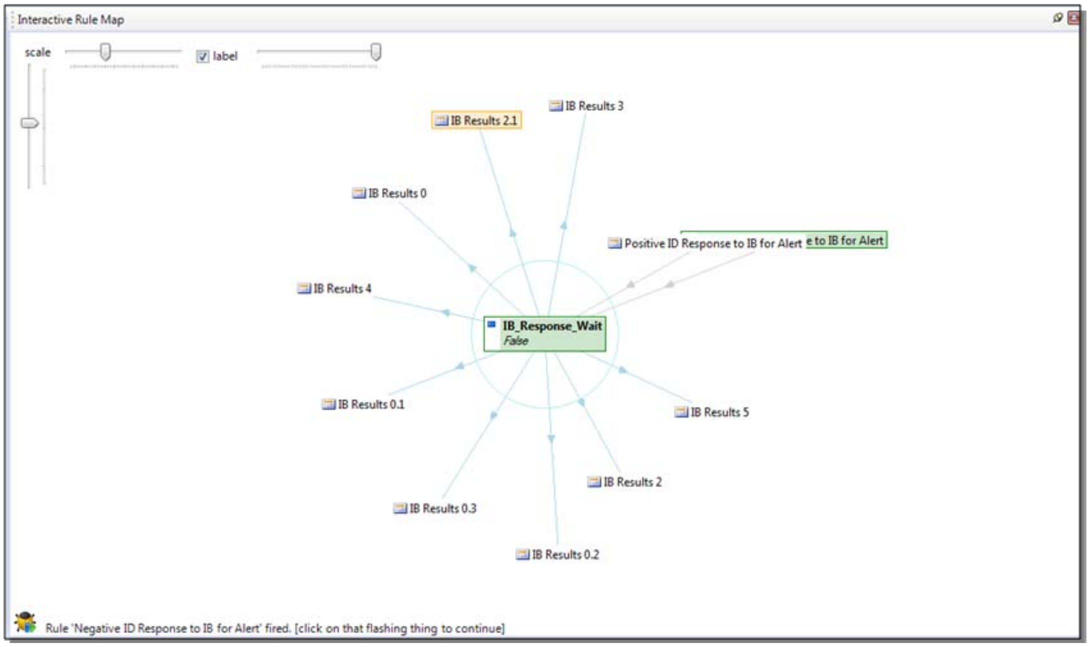
Rule 'Alert Notification is Absent' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Alert Notification is Present
Alert Notification is Absent
Negative ID Response to IB for Alert
Query_Requires_Data_Insertion
False
Positive ID Response to IB for Alert

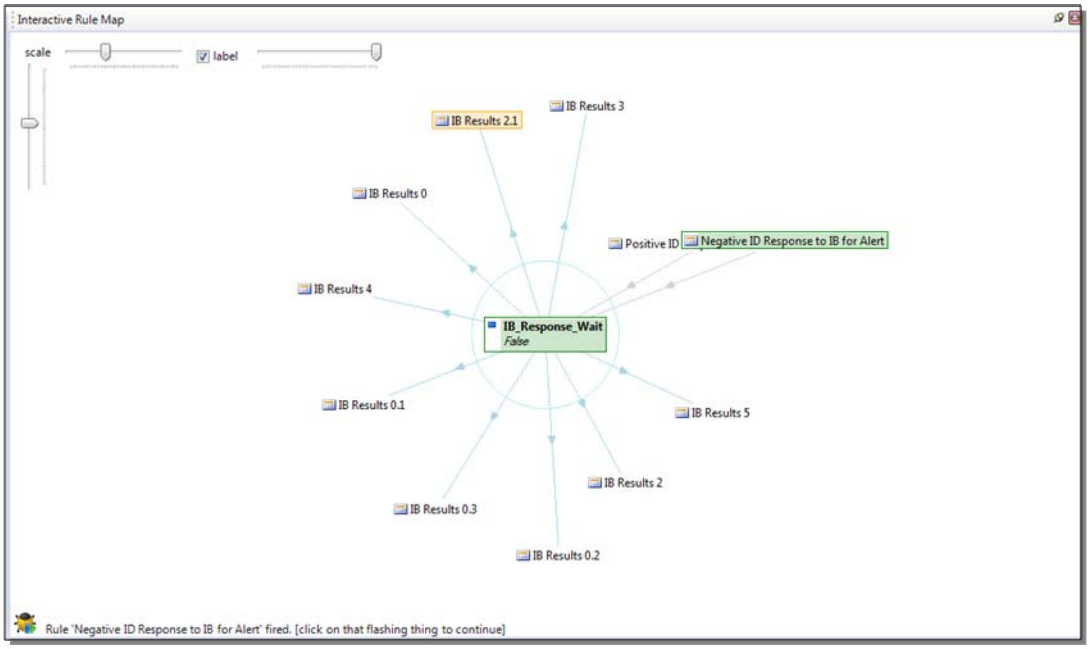
Rule 'Alert Notification is Present' did not fire. [click on that flashing thing to continue]



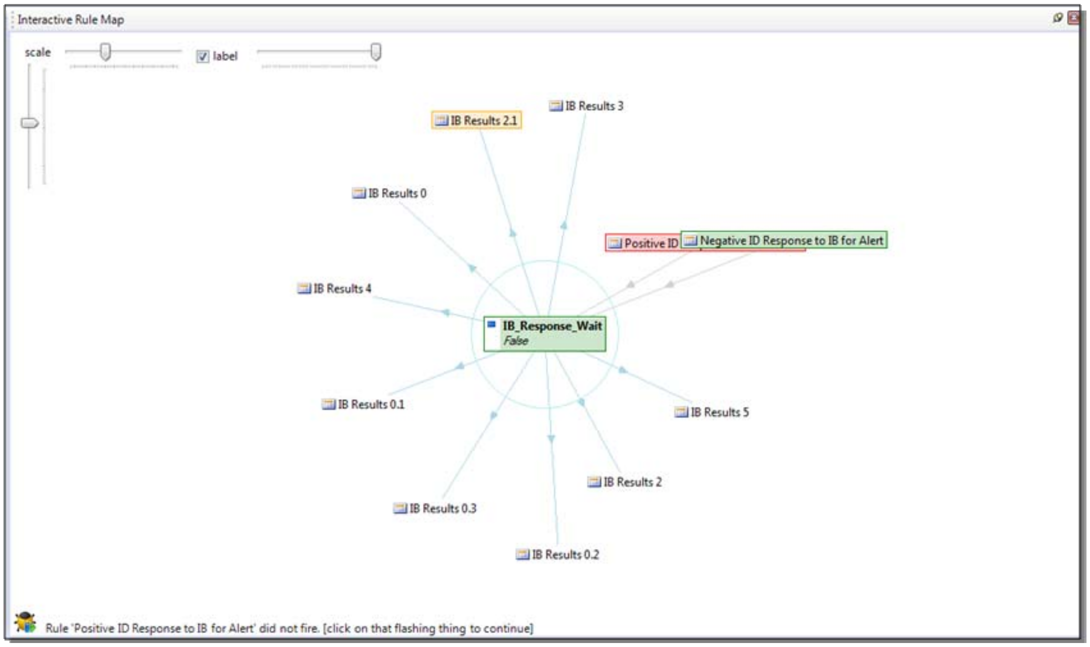
Rule 'Alert Notification is Present' did not fire. [click on that flashing thing to continue]



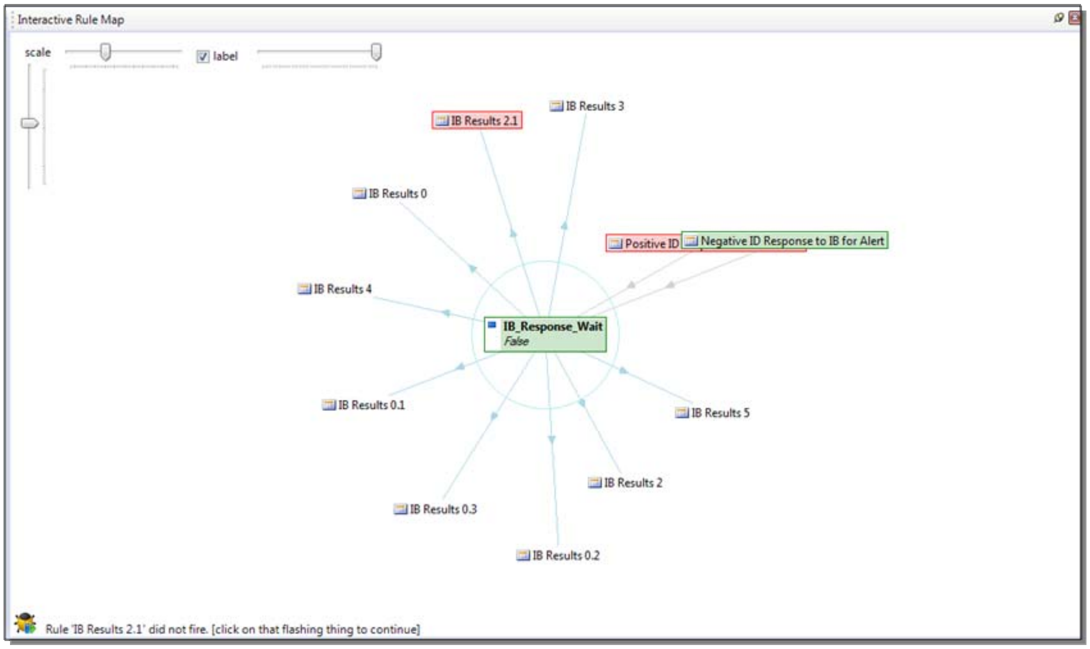
Rule 'Negative ID Response to IB for Alert' fired. [click on that flashing thing to continue]



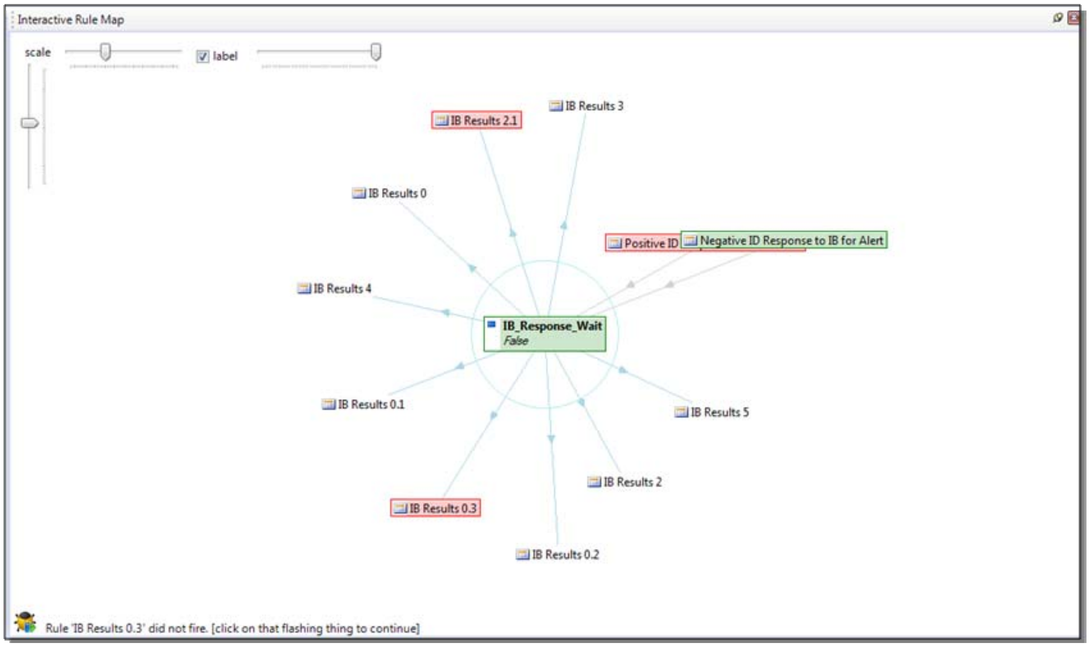
Rule 'Negative ID Response to IB for Alert' fired. [click on that flashing thing to continue]



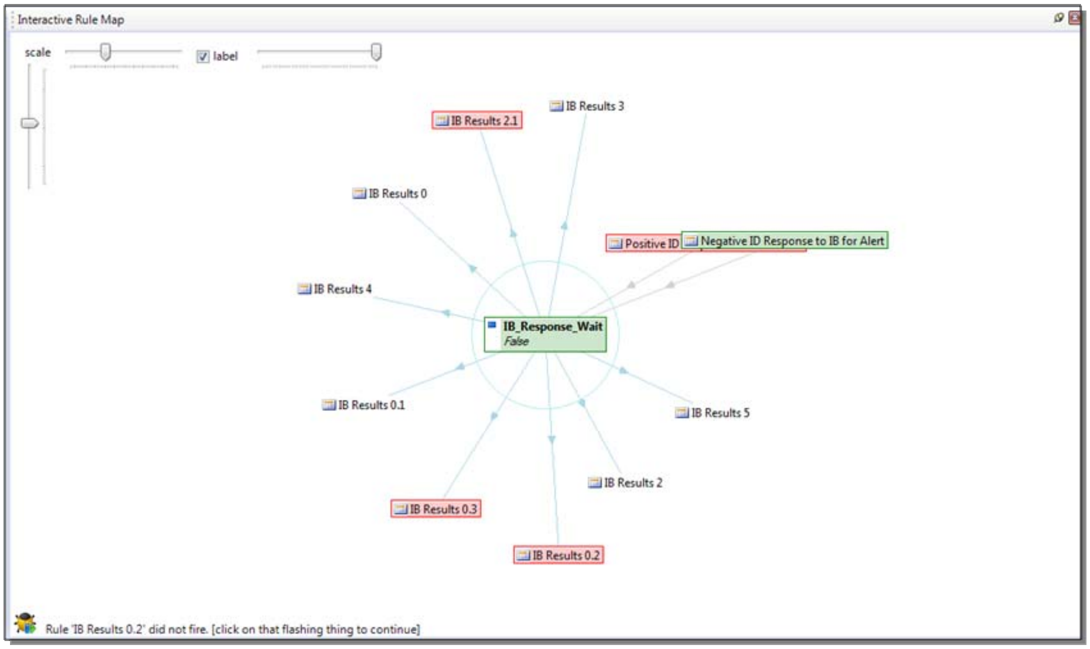
Rule 'Positive ID Response to IB for Alert' did not fire. [click on that flashing thing to continue]



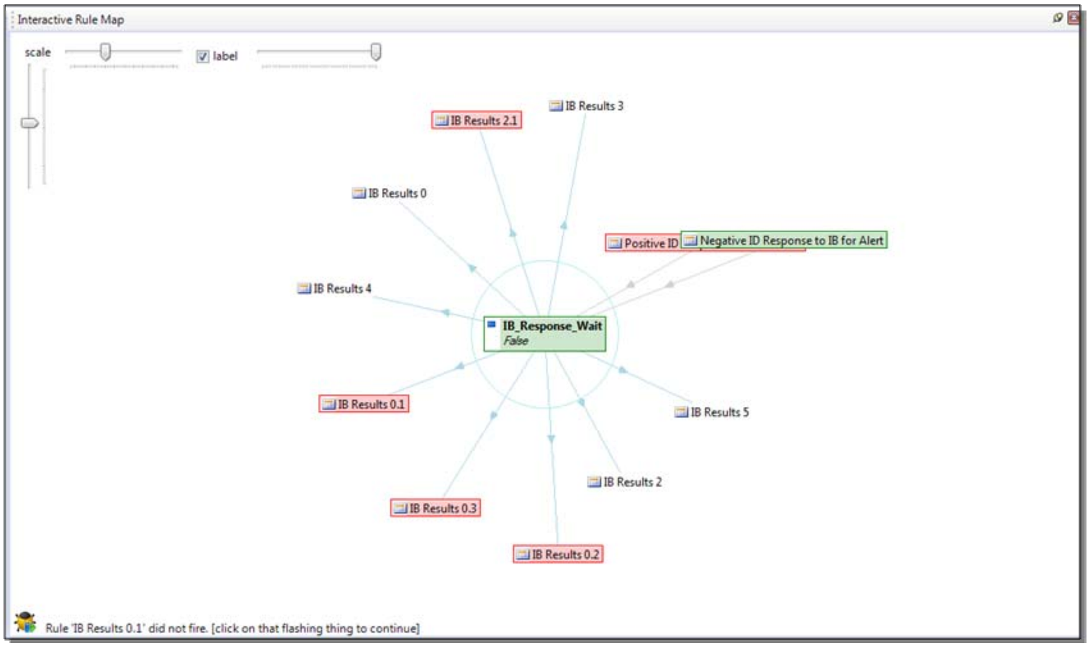
Rule 'IB Results 2.1' did not fire. [click on that flashing thing to continue]



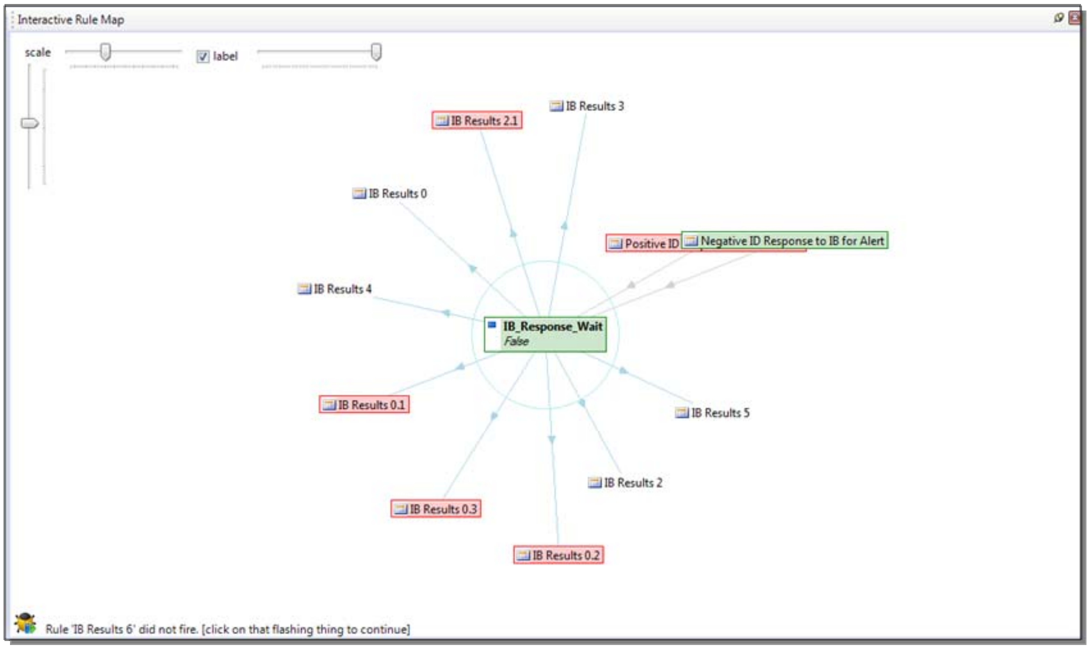
Rule 'IB Results 0.3' did not fire. [click on that flashing thing to continue]



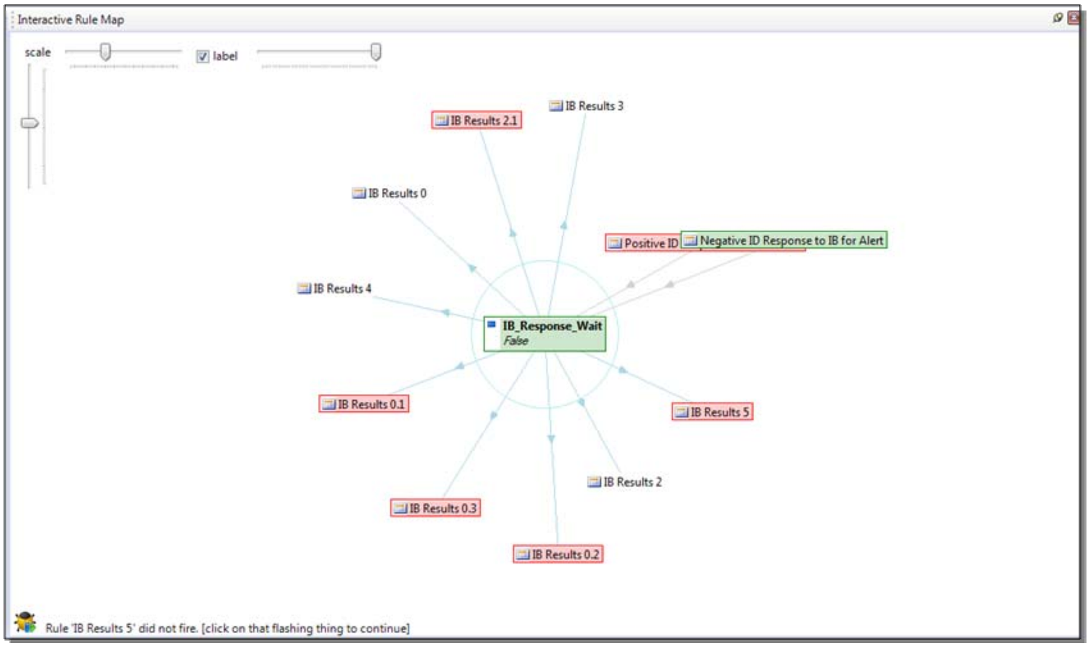
Rule 'IB Results 0.2' did not fire. [click on that flashing thing to continue]



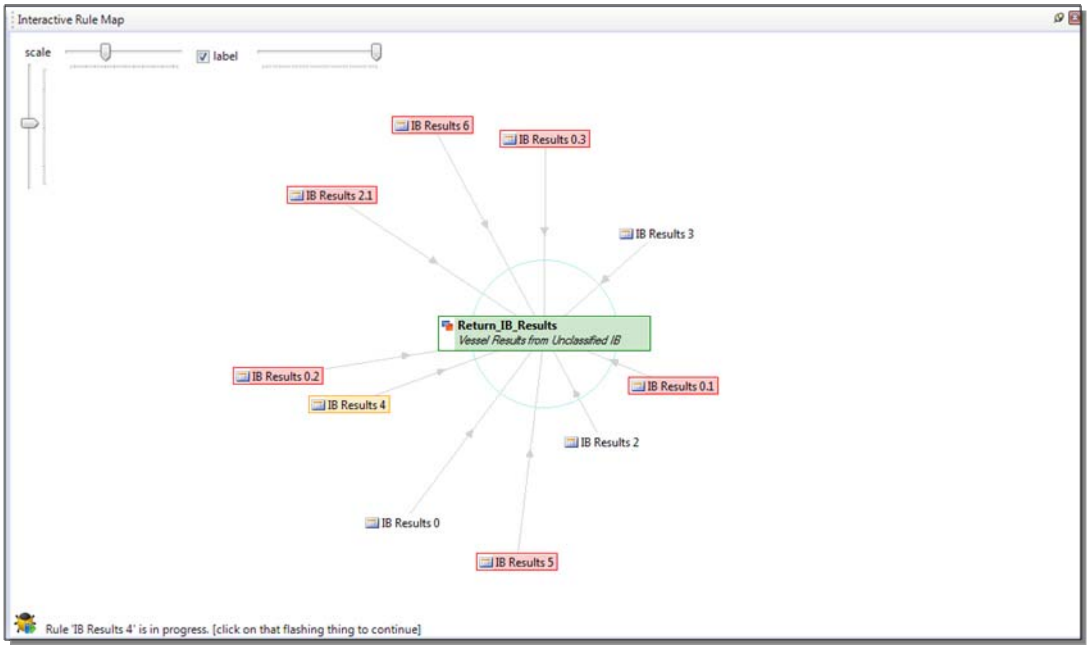
Rule 'IB Results 0.1' did not fire. [click on that flashing thing to continue]



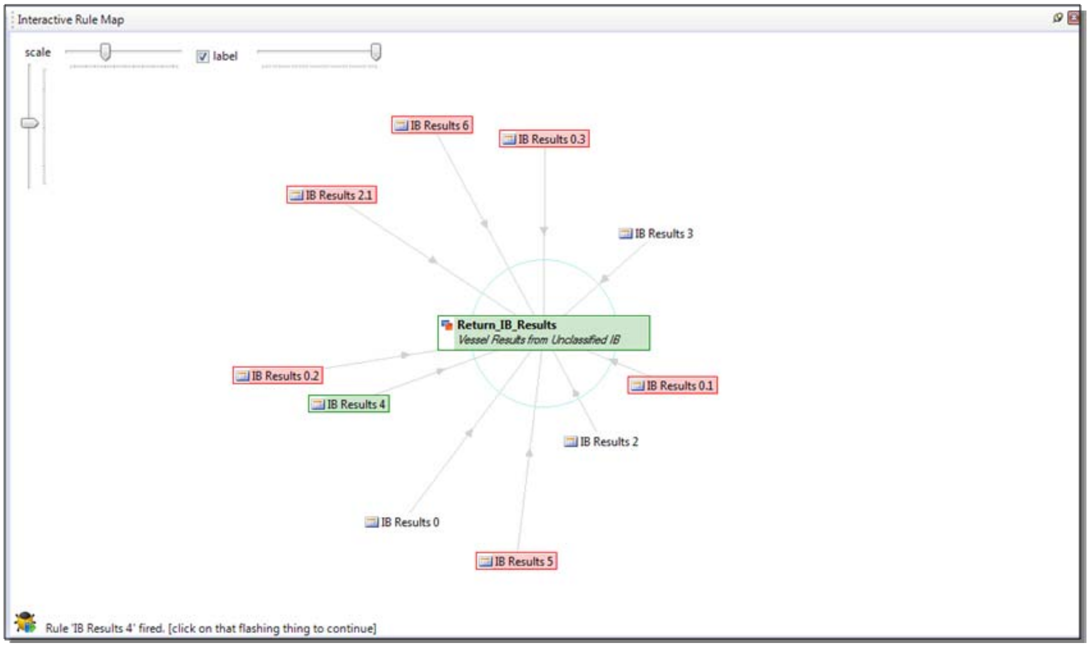
Rule 'IB Results 6' did not fire. [click on that flashing thing to continue]



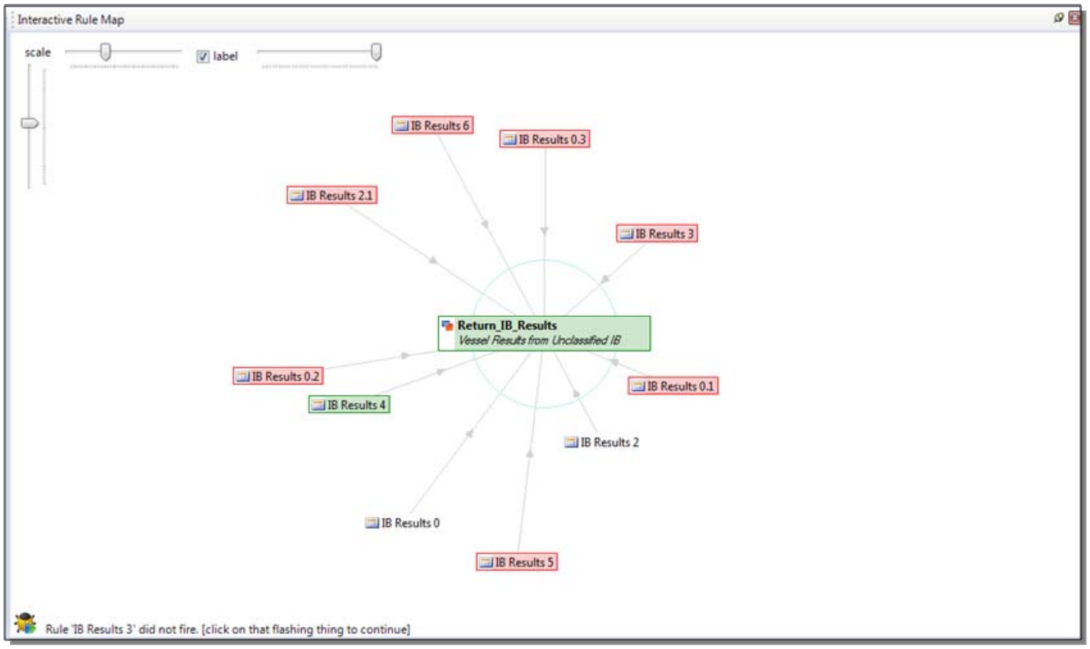
Rule 'IB Results 5' did not fire. [click on that flashing thing to continue]



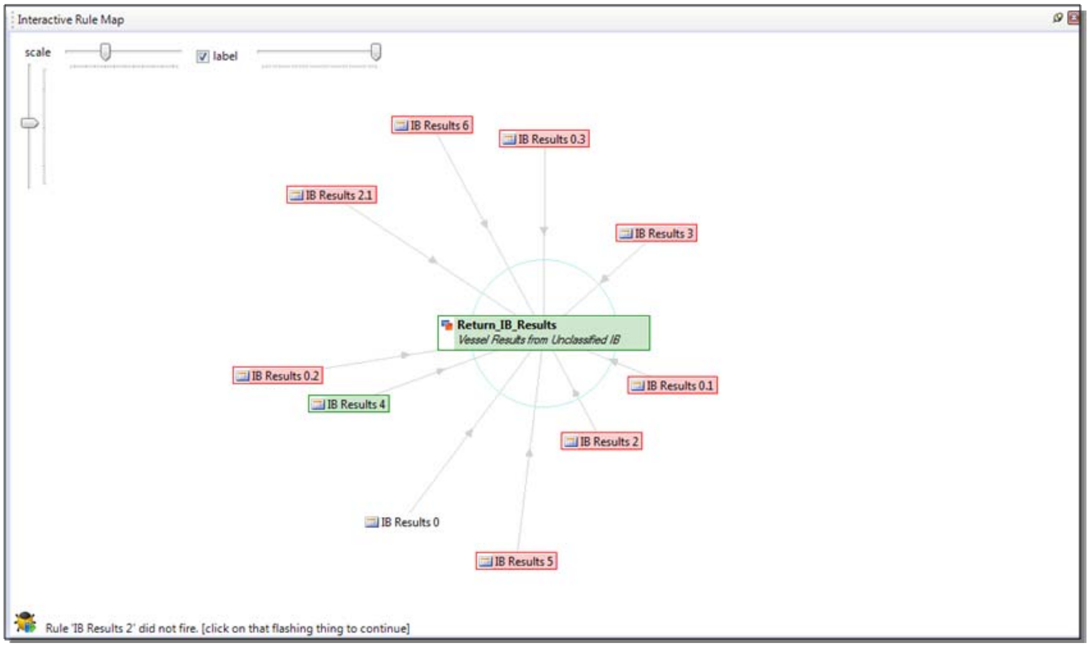
Rule 'IB Results 4' is in progress. [click on that flashing thing to continue]



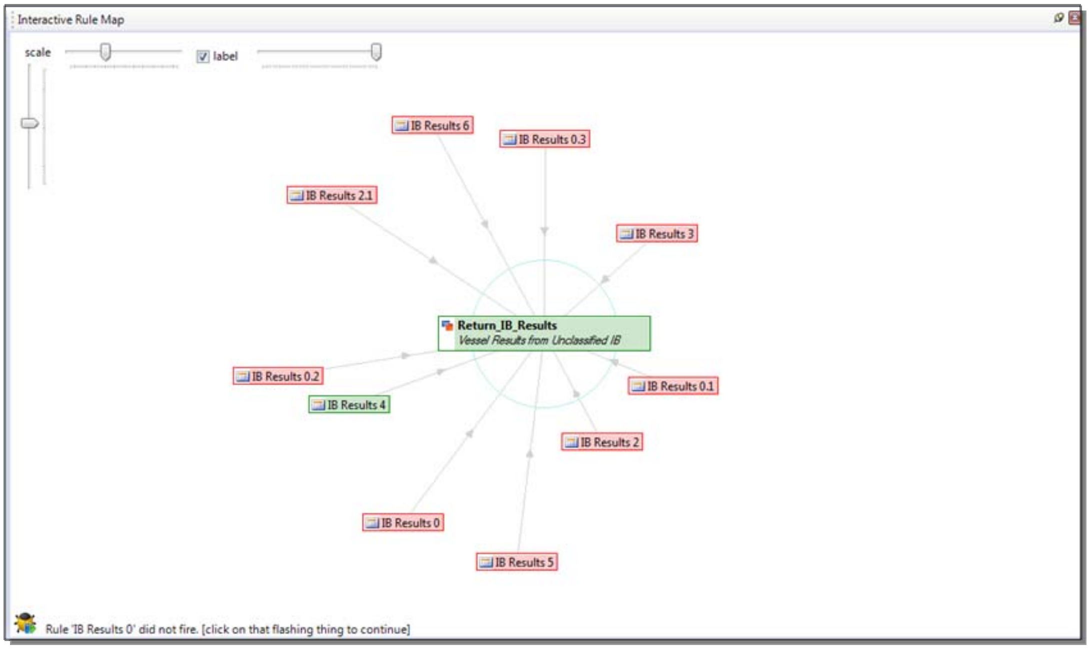
Rule 'IB Results 4' fired. [click on that flashing thing to continue]



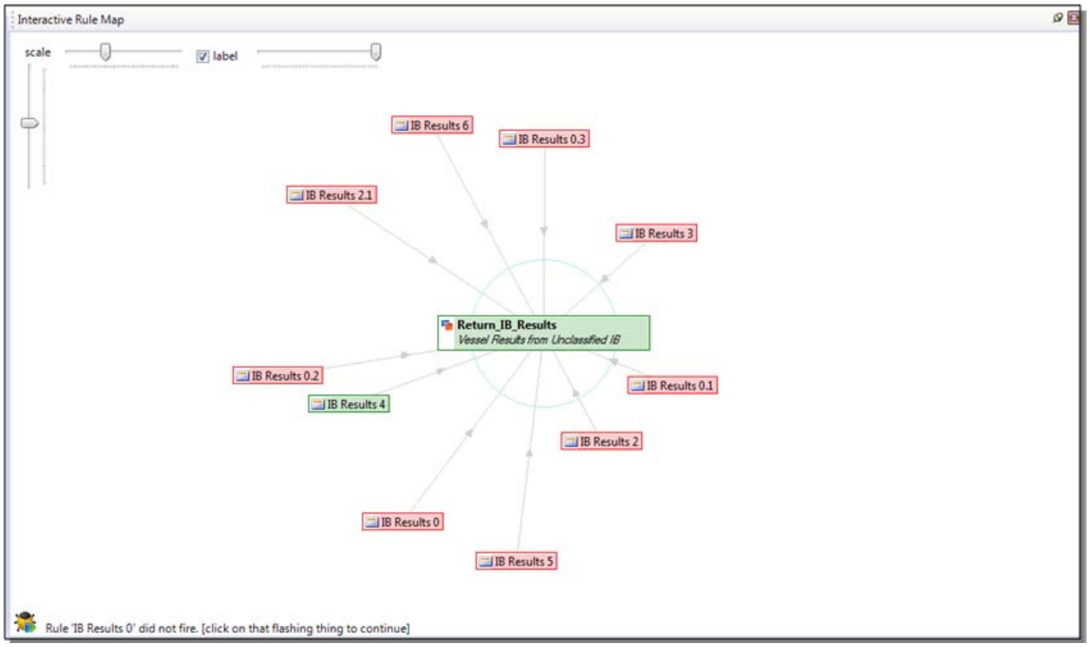
Rule 'IB Results 3' did not fire. [click on that flashing thing to continue]



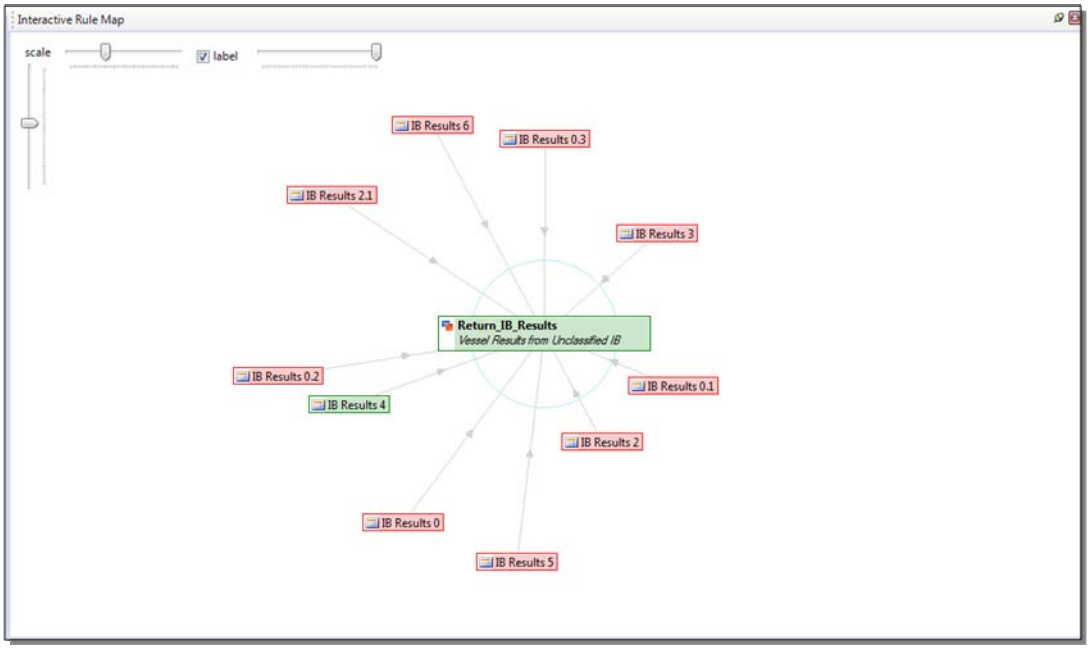
Rule 'IB Results 2' did not fire. [click on that flashing thing to continue]



Rule 'IB Results 0' did not fire. [click on that flashing thing to continue]



Rule 'IB Results 0' did not fire. [click on that flashing thing to continue]



THIS PAGE INTENTIONALLY LEFT BLANK

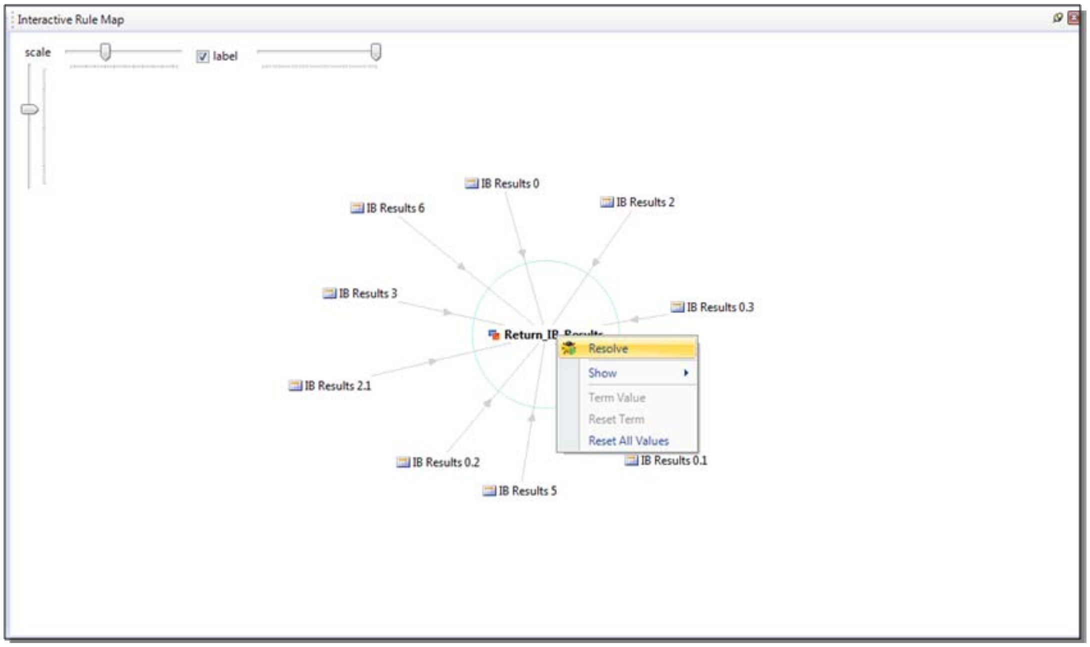
APPENDIX D. RULE TRACE OF MISUSE CASE PREVENTED

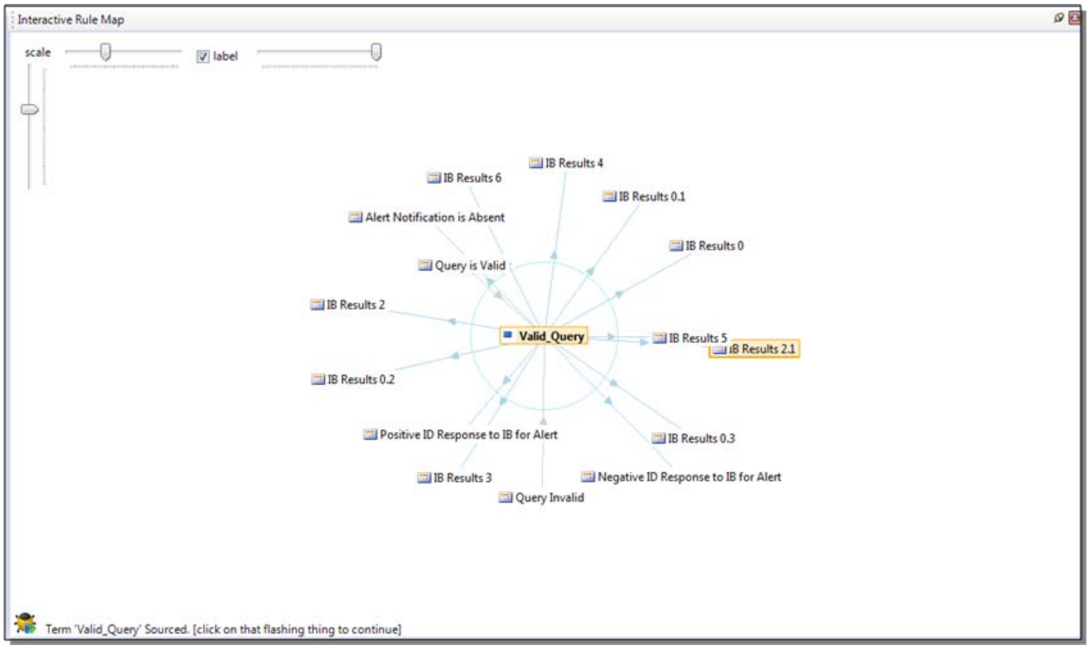
The visual trace of the ruleset execution was used to show that the misuse highlighted by the Misuse Case and accounted for in the Security Use Case is prevented through the RuleML ruleset. This trace is completed using the following parameters:

Type of query:	Destination Port
Role / Actor:	Harbormaster
Location:	San Diego
Security Level:	Unclassified
Alert Present:	True
Alert Classification Level:	Secret
Expected Result:	“Vessel Results from Unclassified IB” with ID Injection for Port of San Diego”

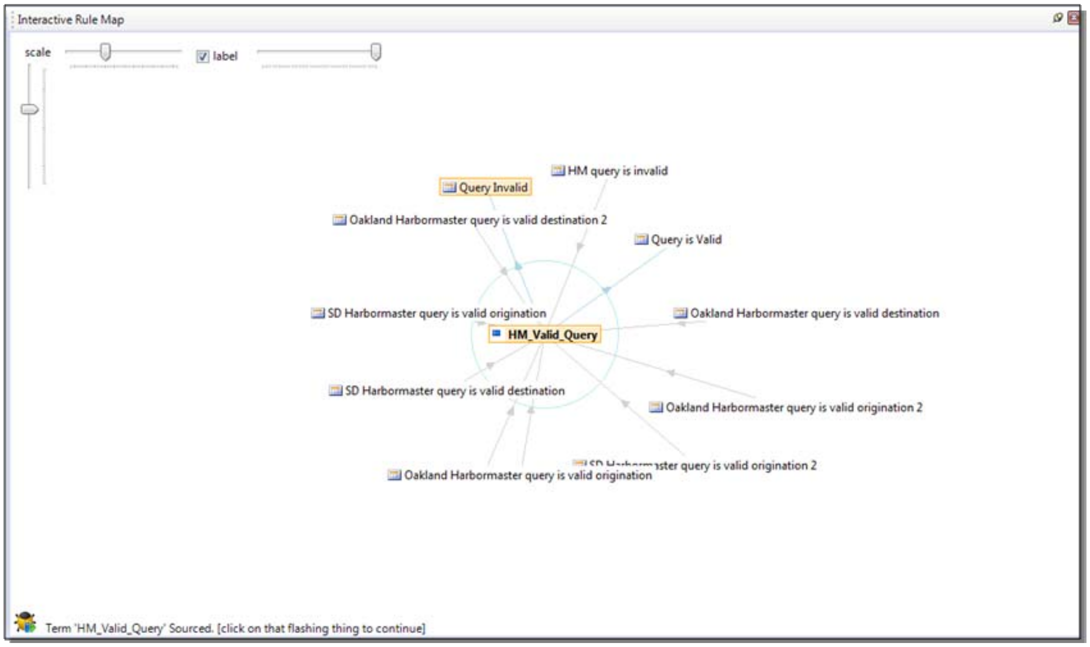
The expected result from this query and the RuleML execution is an IB response of “Vessel Results from Unclassified IB with ID Injection for Port of San Diego” to the user. The trace was completed using RuleManager’s Interactive Rule Map functionality. The pop-up boxes shown throughout the trace indicate the sourcing of predicates that would be included with tagged data in a live system. This was not replicated for this research and instead was manually inserted via the dialog boxes.

For the rule trace shown and from the interactive rule map of RuleManager, various indicators are used to show the actions during the trace. A rule or variable highlighted in Yellow, indicates that a rule or variable is currently being sourced. A rule depicted in Red indicates that the rule did not fire (execute) from the ruleset. A rule shown highlighted in Green indicates that the rule did fire and the result of that firing is also shown in the green highlighted box with the predicate name. Pop-up dialog boxes shown in the trace indicate the sourcing of a variable or predicate that would be done by an external entity (service) or taken from XML attributes attached to the query upon origination (i.e., user role and user security level).

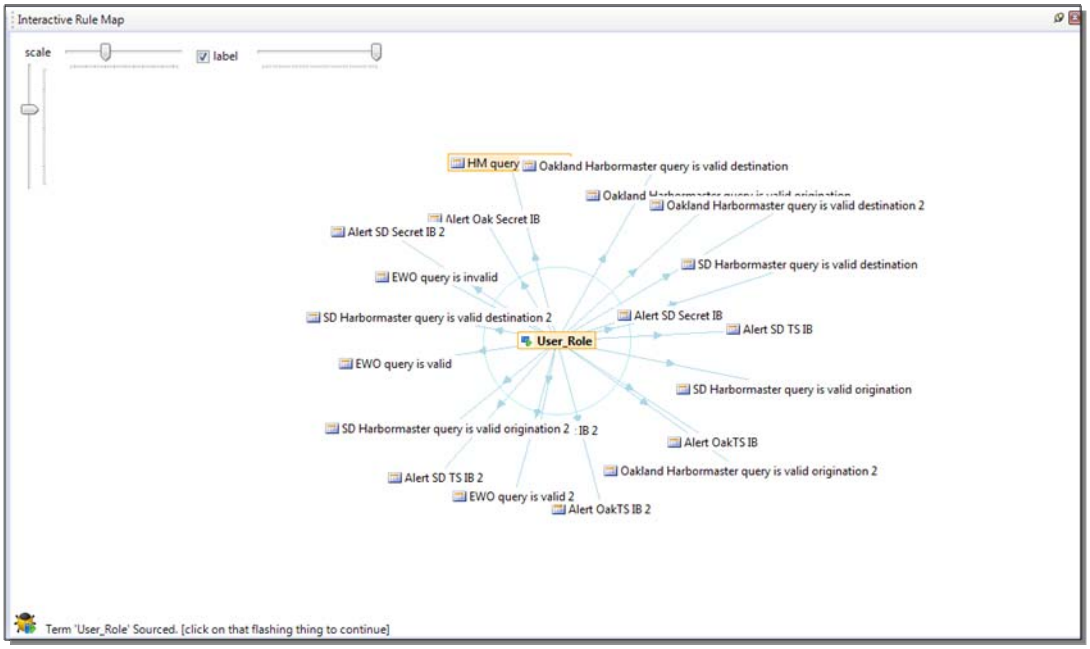




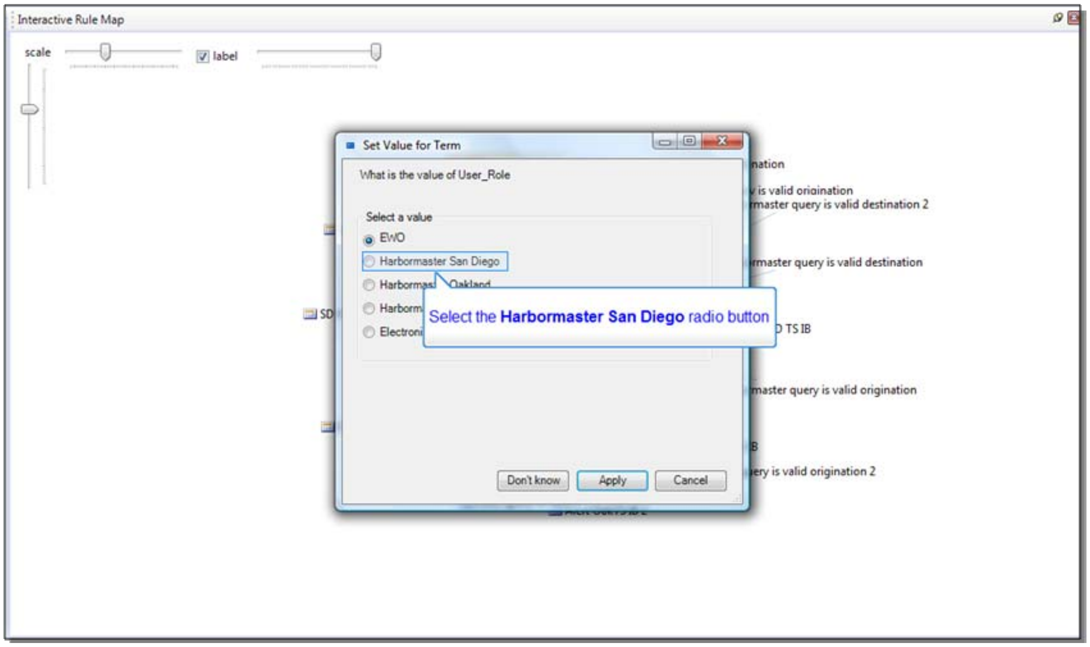
Term 'Valid_Query' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
HM query is invalid
Query Invalid
Oakland Harbormaster query is valid destination 2
Query is Valid
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid destination
HM_Valid_Query
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination

Term 'HM_Valid_Query' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
HM query
Oakland Harbormaster query is valid destination
Oakland Harbormaster query is valid origination
Oakland Harbormaster query is valid destination 2
Alert Oak Secret IB
Alert SD Secret IB 2
SD Harbormaster query is valid destination
EWO query is invalid
SD Harbormaster query is valid destination 2
Alert SD Secret IB
Alert SD TS IB
User_Role
EWO query is valid
SD Harbormaster query is valid origination
SD Harbormaster query is valid origination 2
IB 2
Alert OakTS IB
Oakland Harbormaster query is valid origination 2
Alert SD TS IB 2
EWO query is valid 2
Alert OakTS IB 2

Term 'User_Role' Sourced. [click on that flashing thing to continue]



Interactive Rule Map

scale ☑ label

Set Value for Term

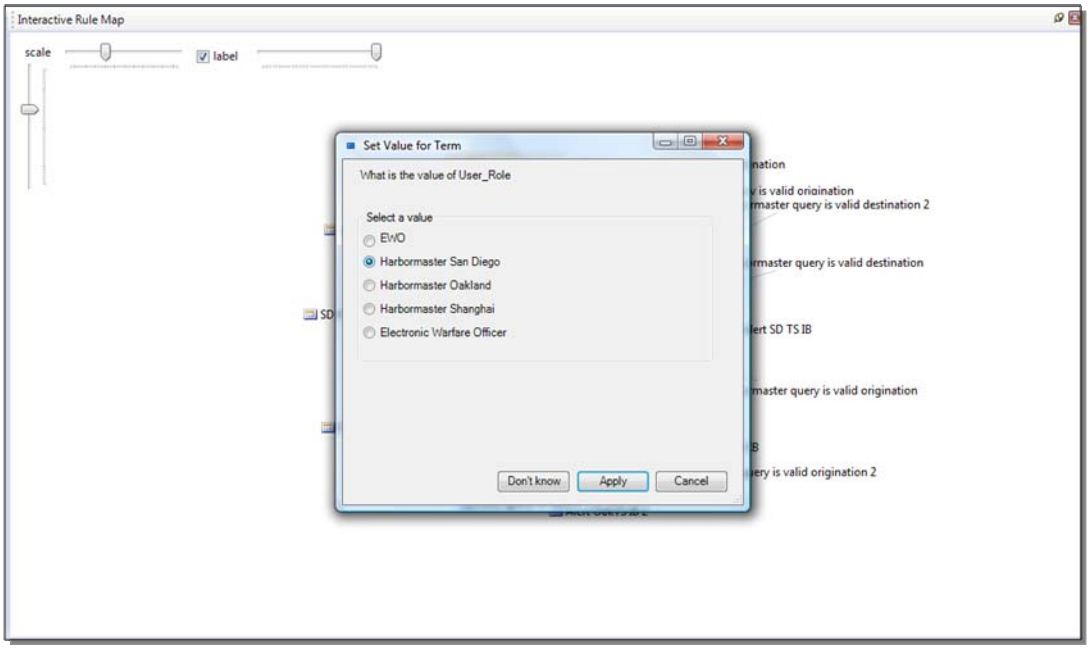
What is the value of User_Role

Select a value

● EWO
○ Harbormaster San Diego
○ Harbormas Oakland
○ Harborm
○ Electroni

Select the **Harbormaster San Diego** radio button

nation
y is valid origination
rmaster query is valid destination 2
rmaster query is valid destination
SD
D TS IB
master query is valid origination
B
ery is valid origination 2
Don't know
Apply
Cancel



Interactive Rule Map

scale ☑ label

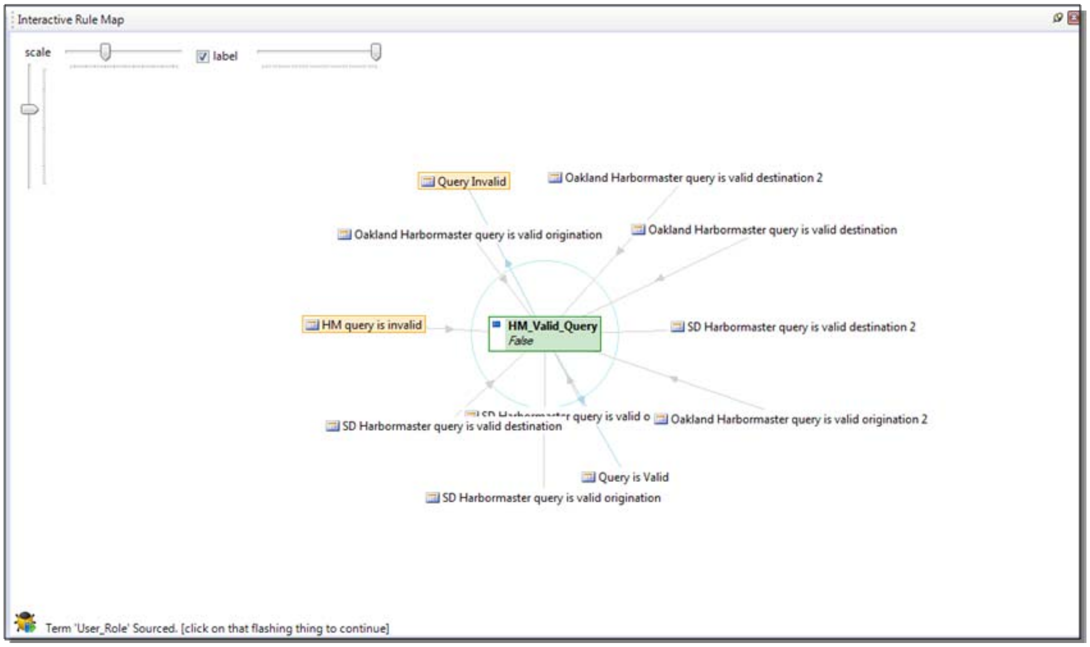
Set Value for Term

What is the value of User_Role

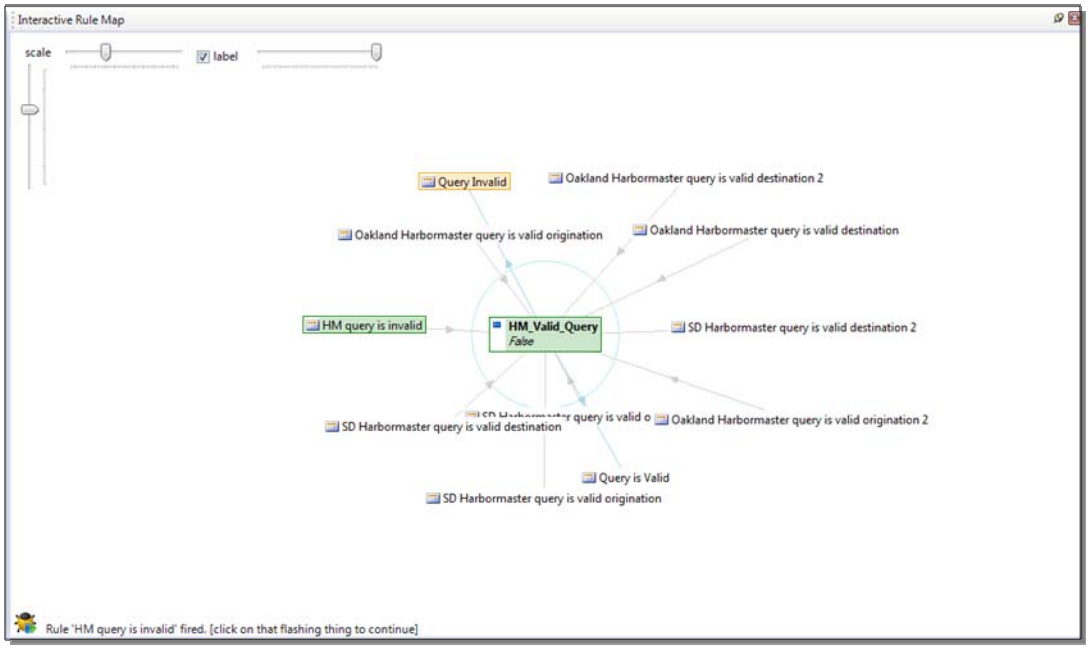
Select a value

○ EWO
● Harbormaster San Diego
○ Harbormaster Oakland
○ Harbormaster Shanghai
○ Electronic Warfare Officer

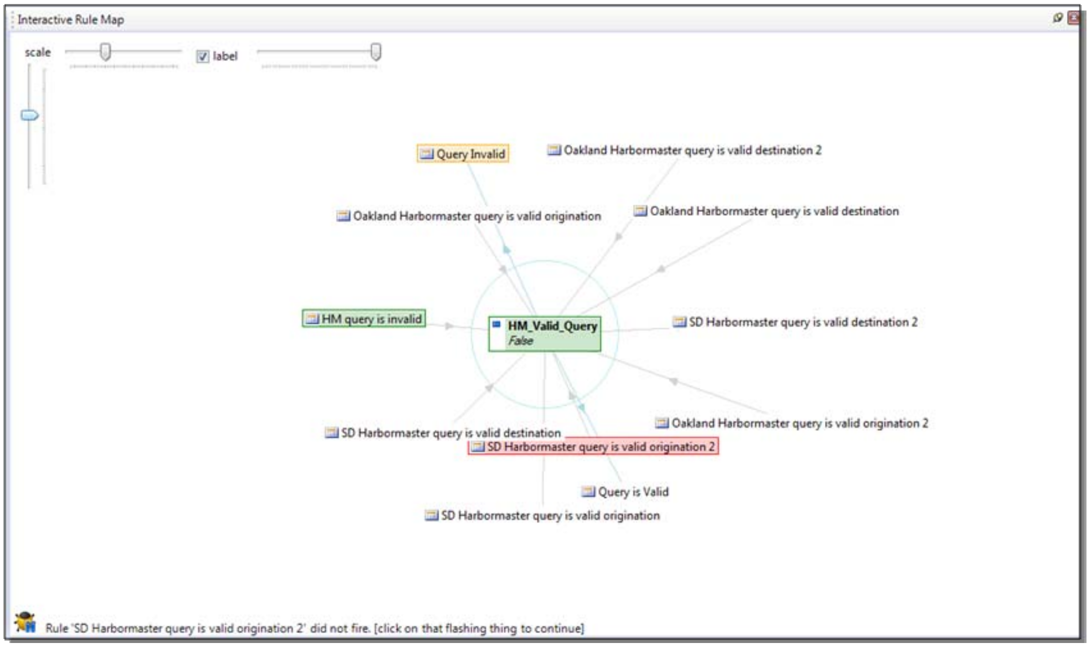
nation
y is valid origination
rmaster query is valid destination 2
rmaster query is valid destination
SD
lert SD TS IB
master query is valid origination
B
ery is valid origination 2
Don't know
Apply
Cancel


Interactive Rule Map
scale
☑ label
Query Invalid
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination
Oakland Harbormaster query is valid destination
HM query is invalid
HM_Valid_Query
False
SD Harbormaster query is valid destination 2
SD Harbormaster query is valid o
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
Query is Valid
SD Harbormaster query is valid origination

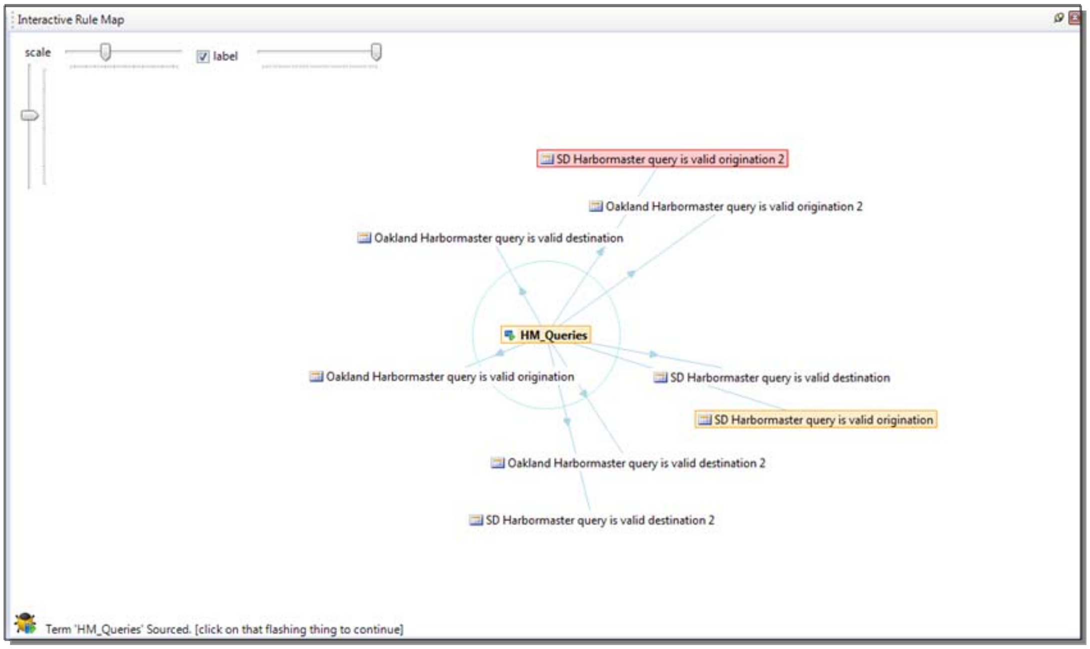
Term 'User_Role' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Query Invalid
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination
Oakland Harbormaster query is valid destination
HM query is invalid
HM_Valid_Query
False
SD Harbormaster query is valid destination 2
SD Harbormaster query is valid o
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
Query is Valid
SD Harbormaster query is valid origination

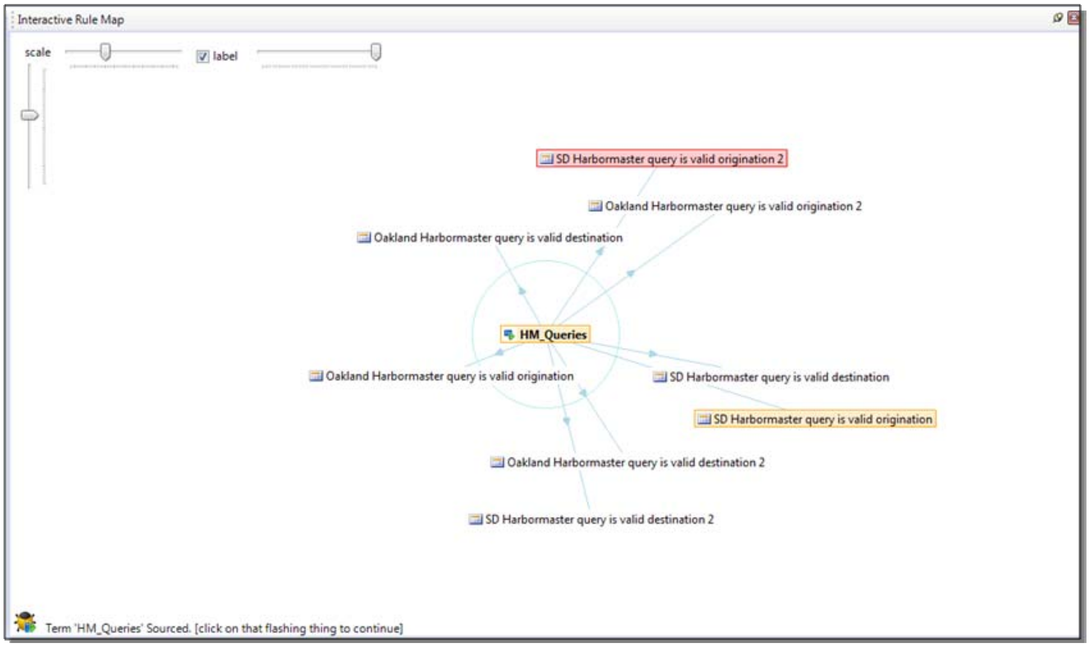
Rule 'HM query is invalid' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Query Invalid
Oakland Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination
Oakland Harbormaster query is valid destination
HM query is invalid
HM_Valid_Query
False
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
SD Harbormaster query is valid origination 2
Query is Valid
SD Harbormaster query is valid origination

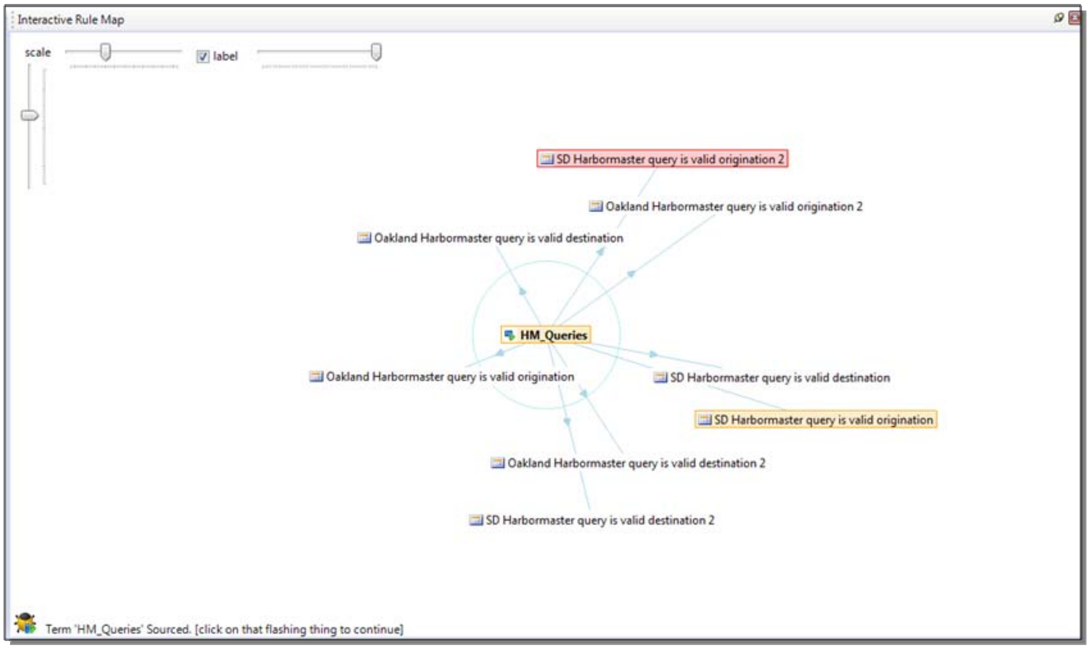
Rule 'SD Harbormaster query is valid origination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination
HM_Queries
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

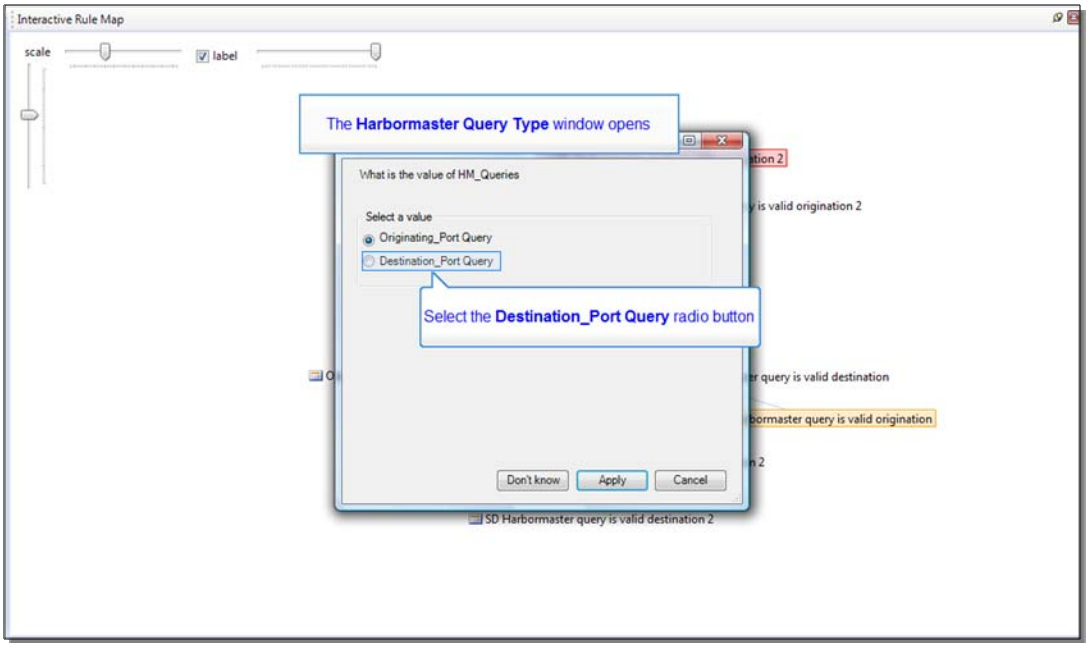
Term 'HM_Queries' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination
HM_Queries
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

Term 'HM_Queries' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination
HM_Queries
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

Term 'HM_Queries' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
label
tion 2
y is valid origination 2
r query is valid destination
bormaster query is valid origination
n 2
SD Harbormaster query is valid destination 2
Don't know
Apply
Cancel

The **Harbormaster Query Type** window opens

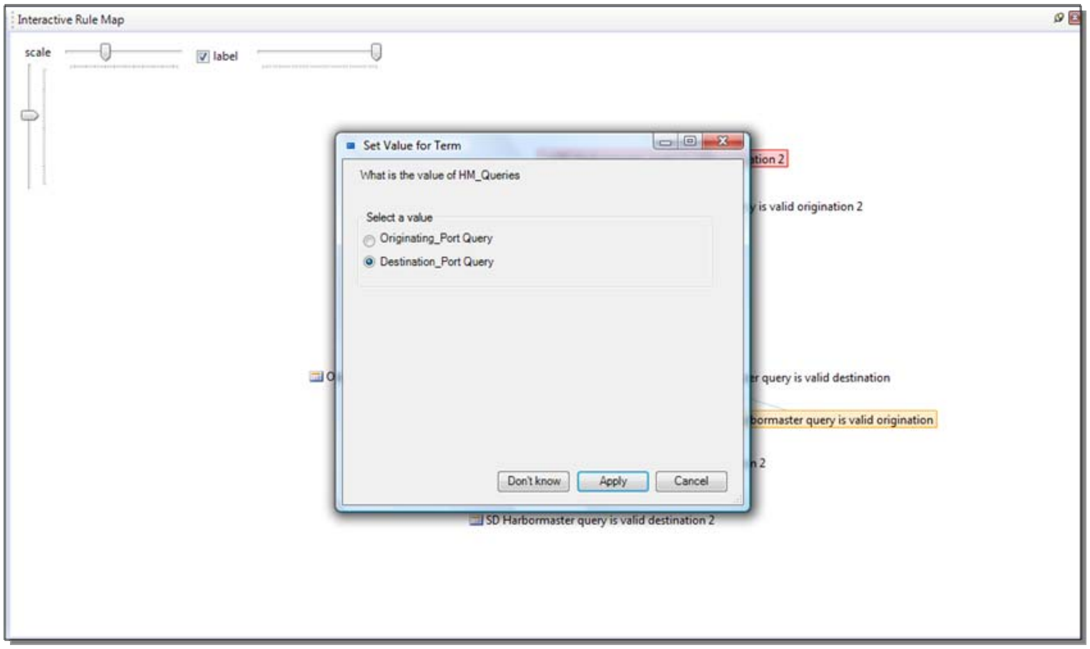
What is the value of HM_Queries

Select a value

● Originating_Port Query

○ Destination_Port Query

Select the **Destination_Port Query** radio button


Interactive Rule Map
scale
label
tion 2
y is valid origination 2
r query is valid destination
bormaster query is valid origination
n 2
SD Harbormaster query is valid destination 2
Don't know
Apply
Cancel

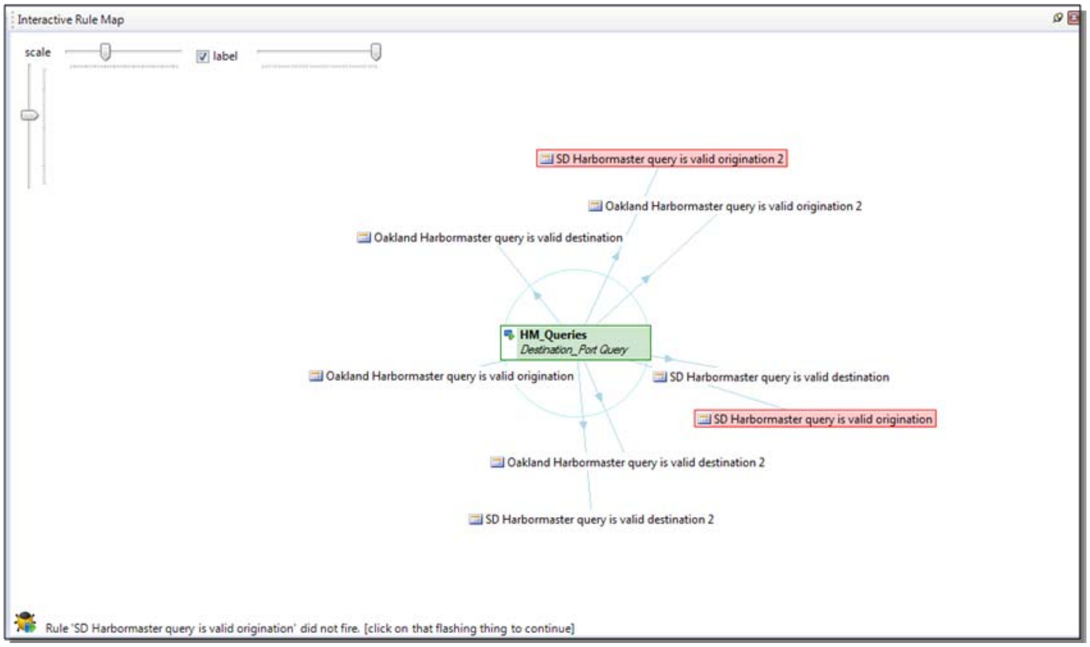
Set Value for Term

What is the value of HM_Queries

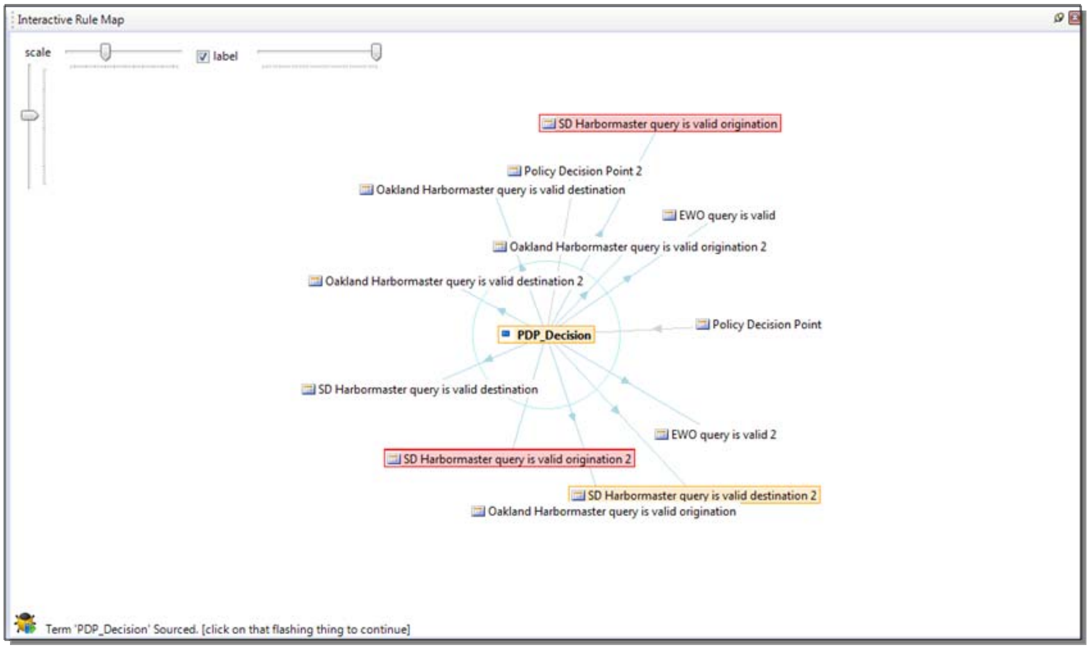
Select a value

○ Originating_Port Query

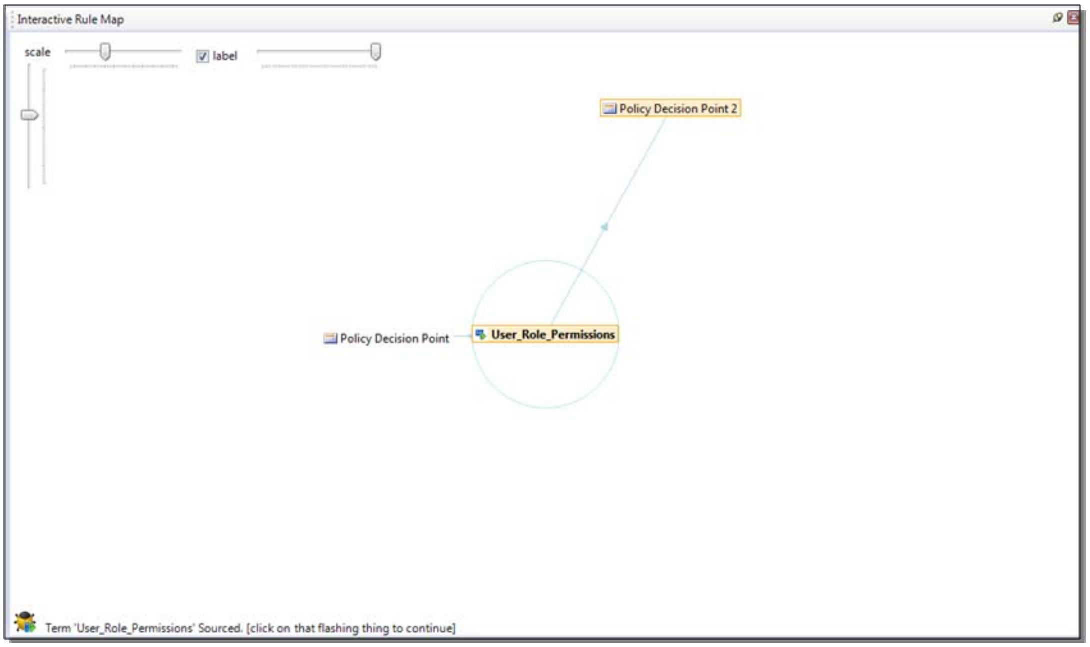
● Destination_Port Query


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination
HM_Queries
Destination_Port Query
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

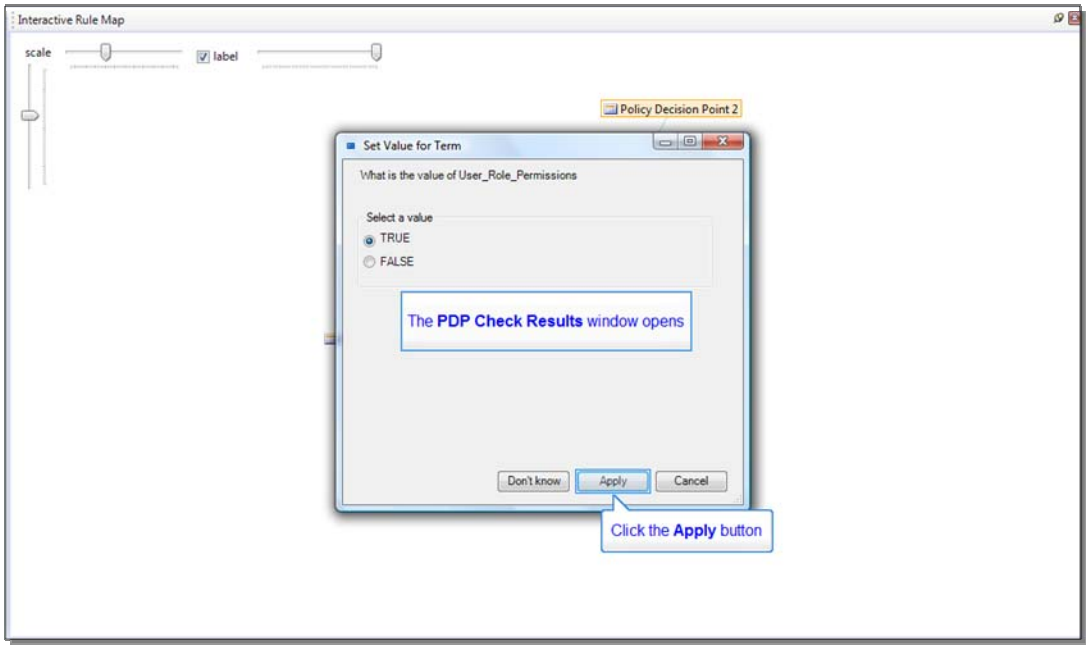
Rule 'SD Harbormaster query is valid origination' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid origination
Policy Decision Point 2
Oakland Harbormaster query is valid destination
EWO query is valid
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination 2
PDP_Decision
Policy Decision Point
SD Harbormaster query is valid destination
EWO query is valid 2
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination

Term 'PDP_Decision' Sourced. [click on that flashing thing to continue]



Term 'User_Role_Permissions' Sourced. [click on that flashing thing to continue]



What is the value of User_Role_Permissions

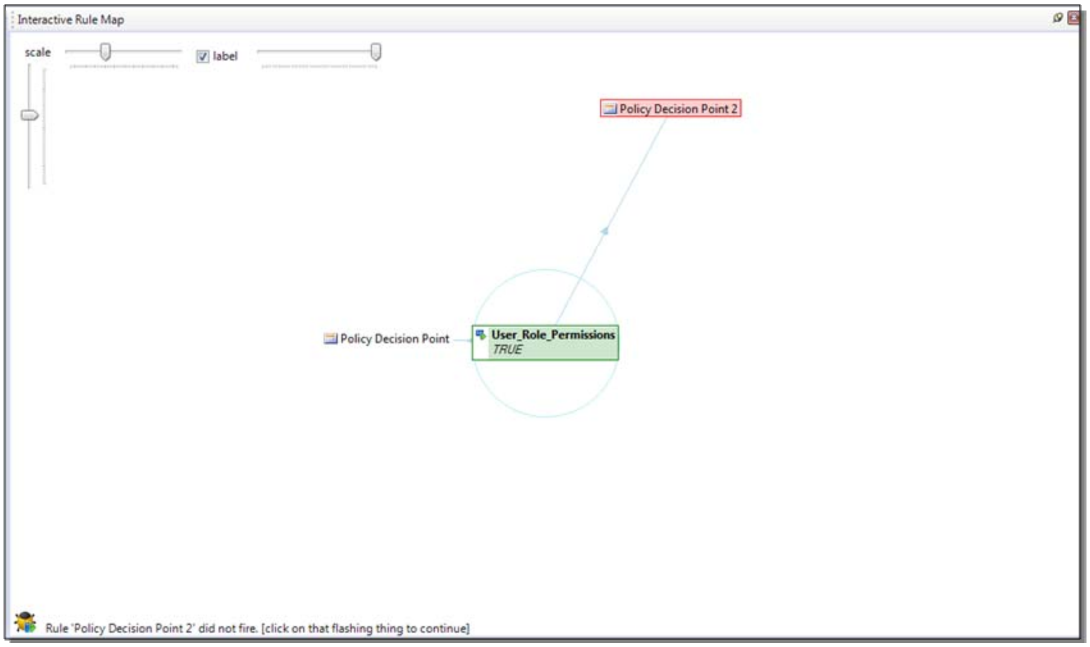
Select a value

● TRUE

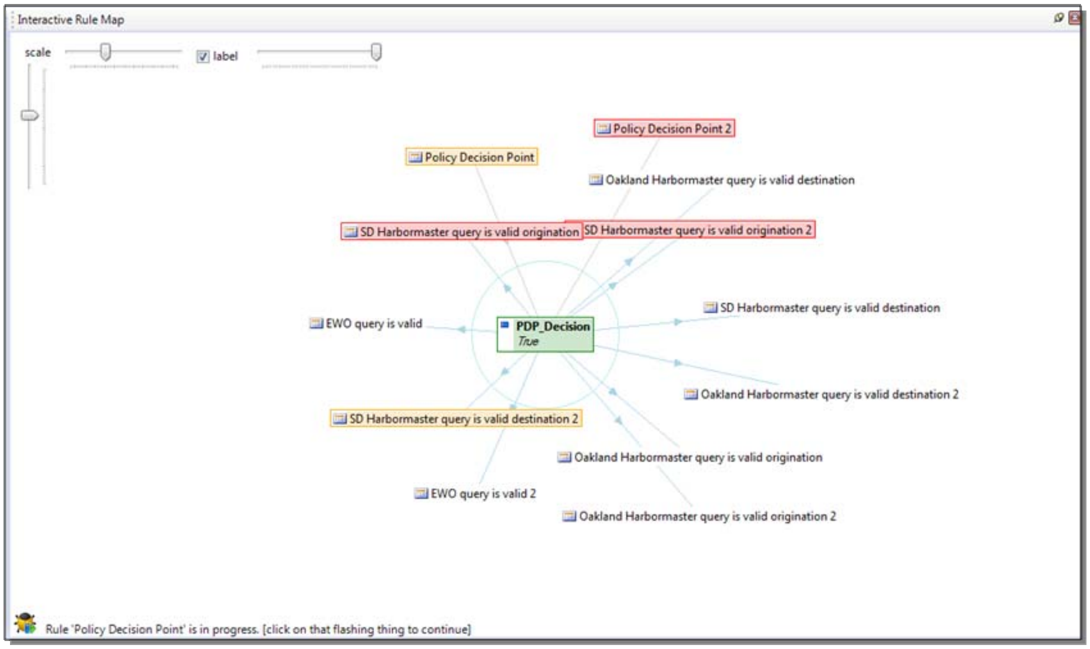
○ FALSE

The **PDP Check Results** window opens

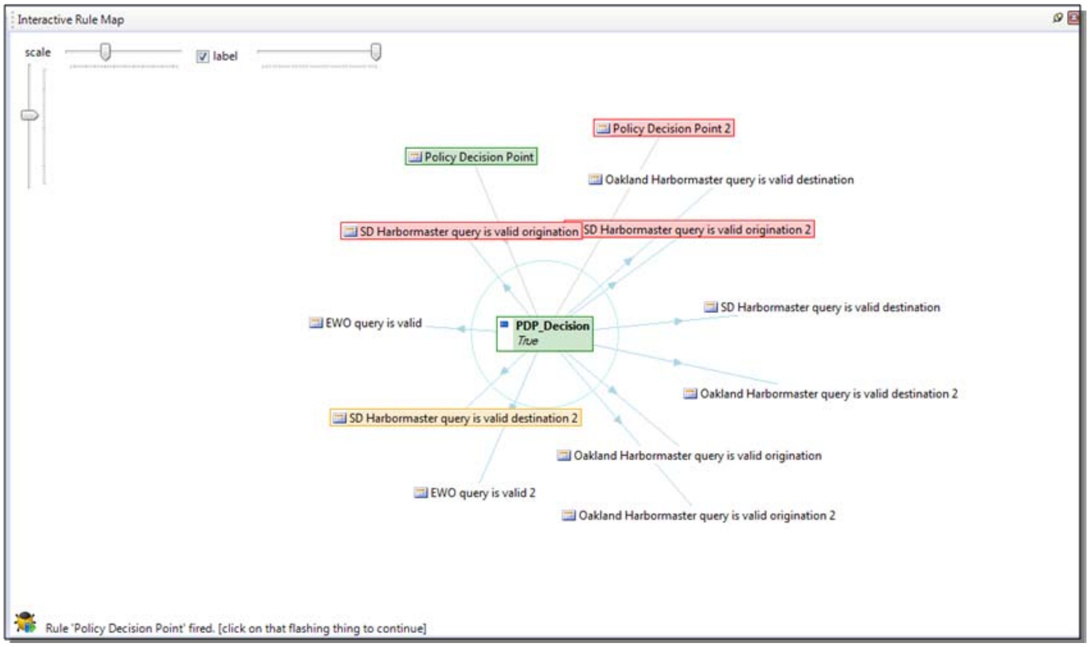
Click the **Apply** button



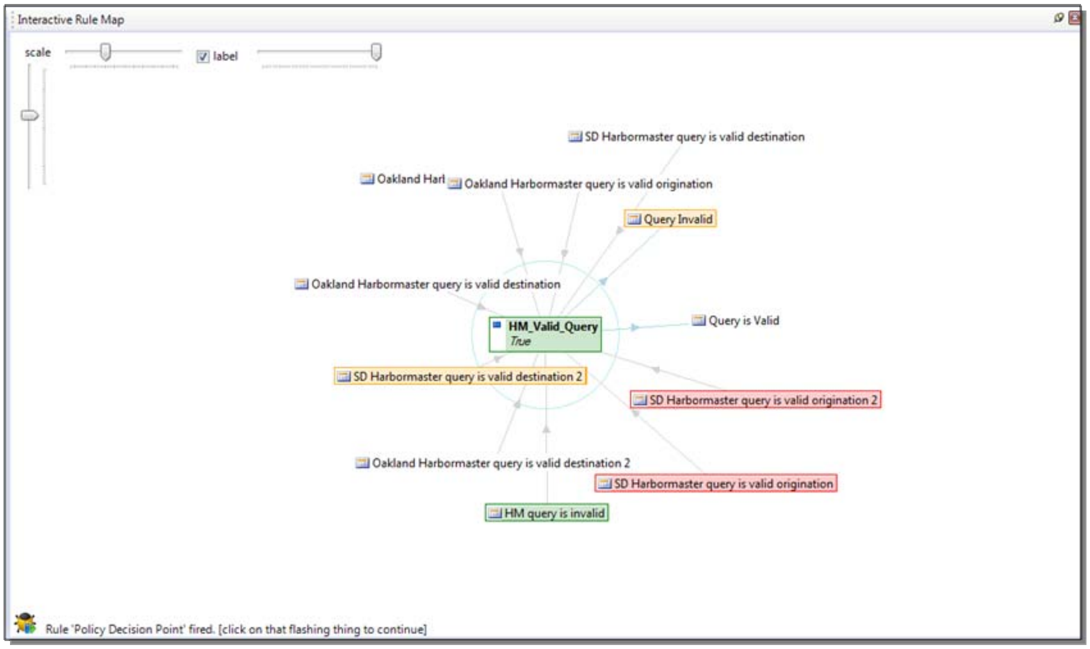
Rule 'Policy Decision Point 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Policy Decision Point 2
Policy Decision Point
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid origination
SD Harbormaster query is valid origination 2
EWO query is valid
PDP_Decision
True
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination
EWO query is valid 2
Oakland Harbormaster query is valid origination 2

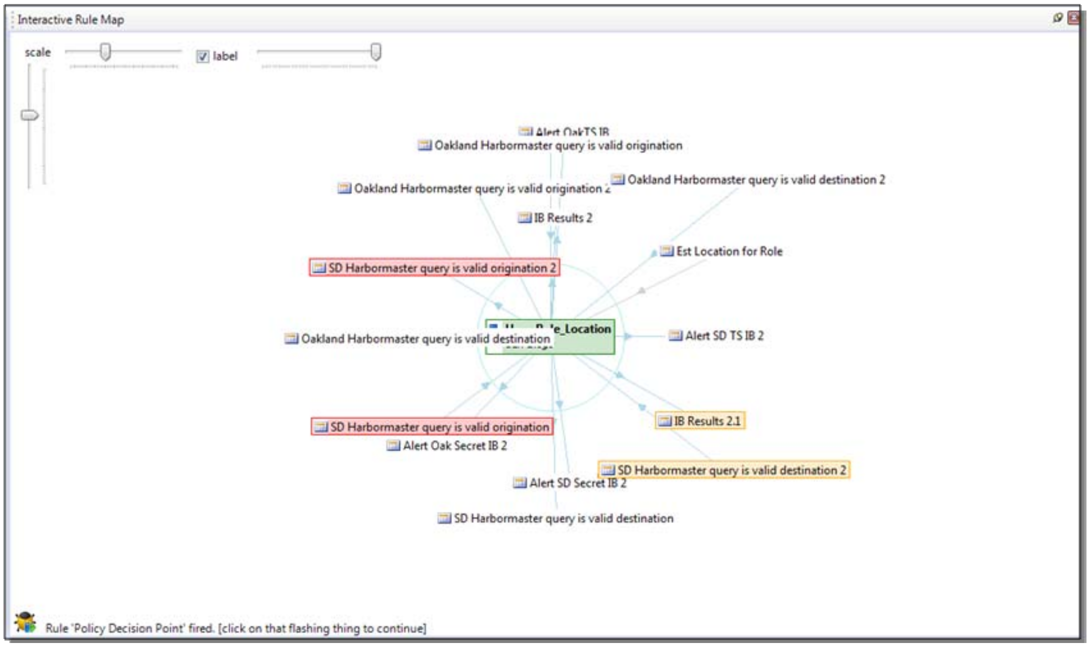
Rule 'Policy Decision Point' is in progress. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Policy Decision Point 2
Policy Decision Point
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid origination
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
EWO query is valid
PDP_Decision
True
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid origination
EWO query is valid 2
Oakland Harbormaster query is valid origination 2

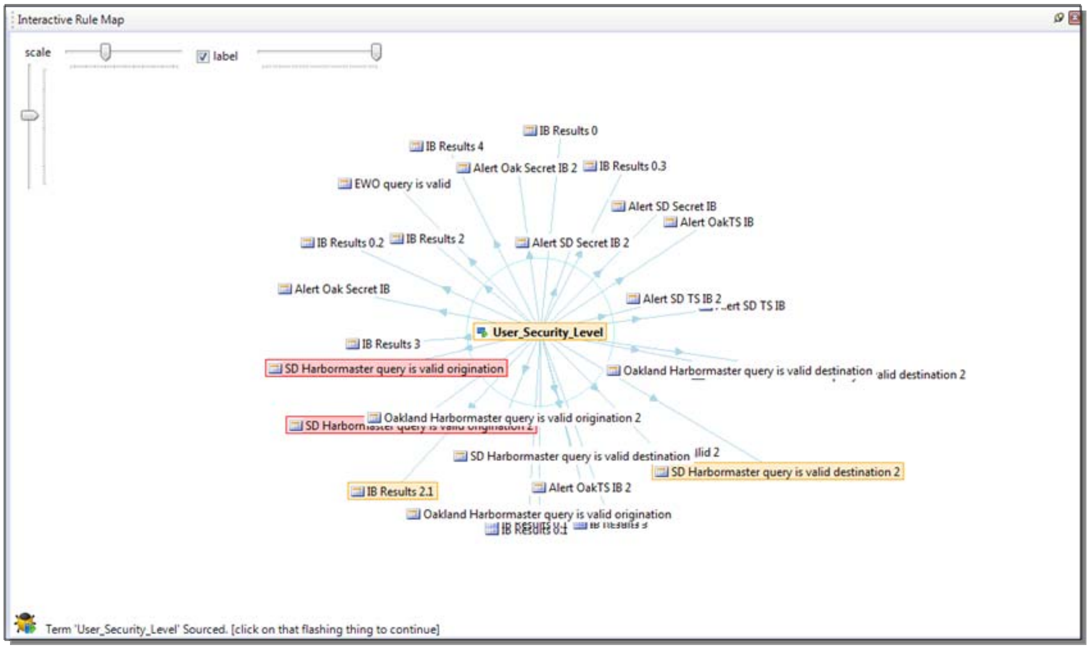
Rule 'Policy Decision Point' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination
Oakland Harl
Oakland Harbormaster query is valid origination
Query Invalid
Oakland Harbormaster query is valid destination
HM_Valid_Query
True
Query is Valid
SD Harbormaster query is valid destination 2
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid origination
HM query is invalid

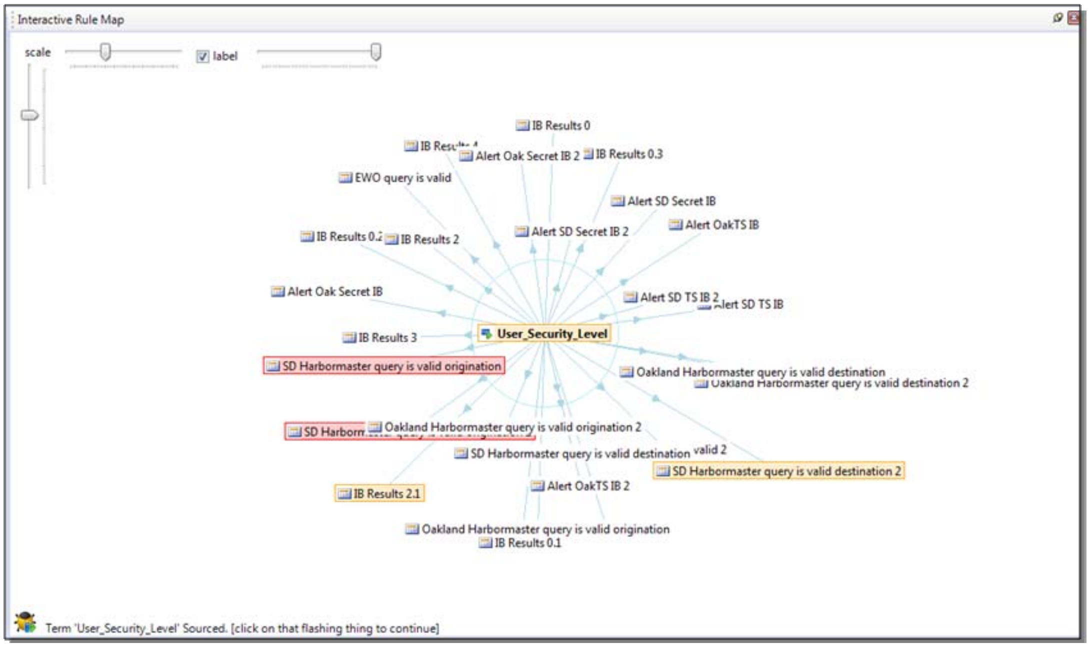
Rule 'Policy Decision Point' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Alert OakTS IB
Oakland Harbormaster query is valid origination
Oakland Harbormaster query is valid origination 2
Oakland Harbormaster query is valid destination 2
IB Results 2
Est Location for Role
SD Harbormaster query is valid origination 2
User_Role_Location
Oakland Harbormaster query is valid destination
Alert SD TS IB 2
SD Harbormaster query is valid origination
IB Results 2.1
Alert Oak Secret IB 2
SD Harbormaster query is valid destination 2
Alert SD Secret IB 2
SD Harbormaster query is valid destination

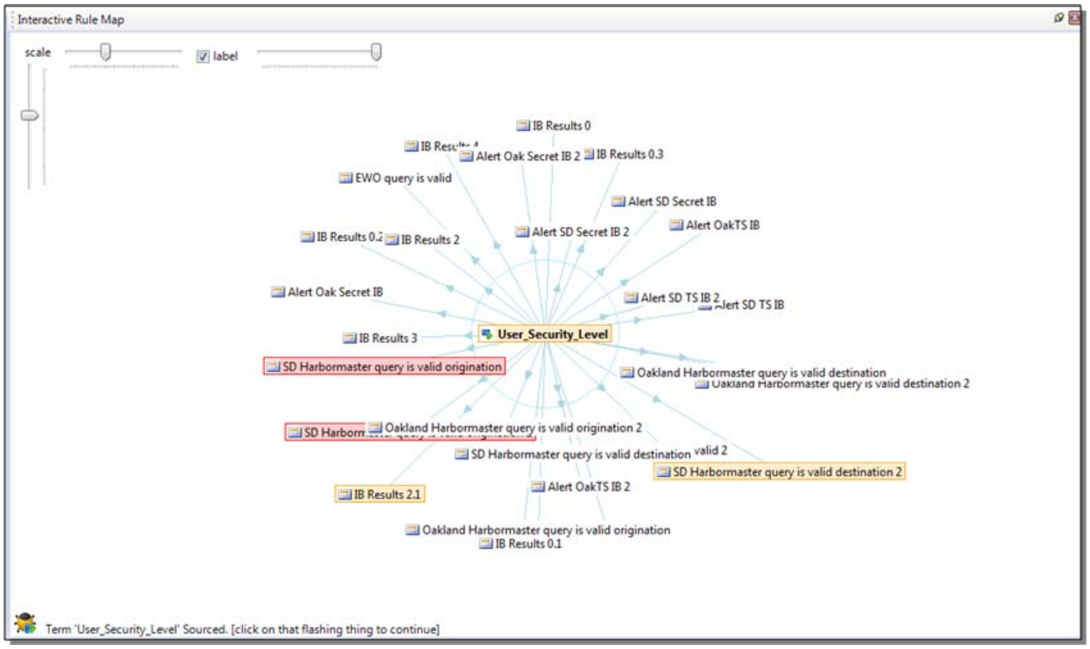
Rule 'Policy Decision Point' fired. [click on that flashing thing to continue]



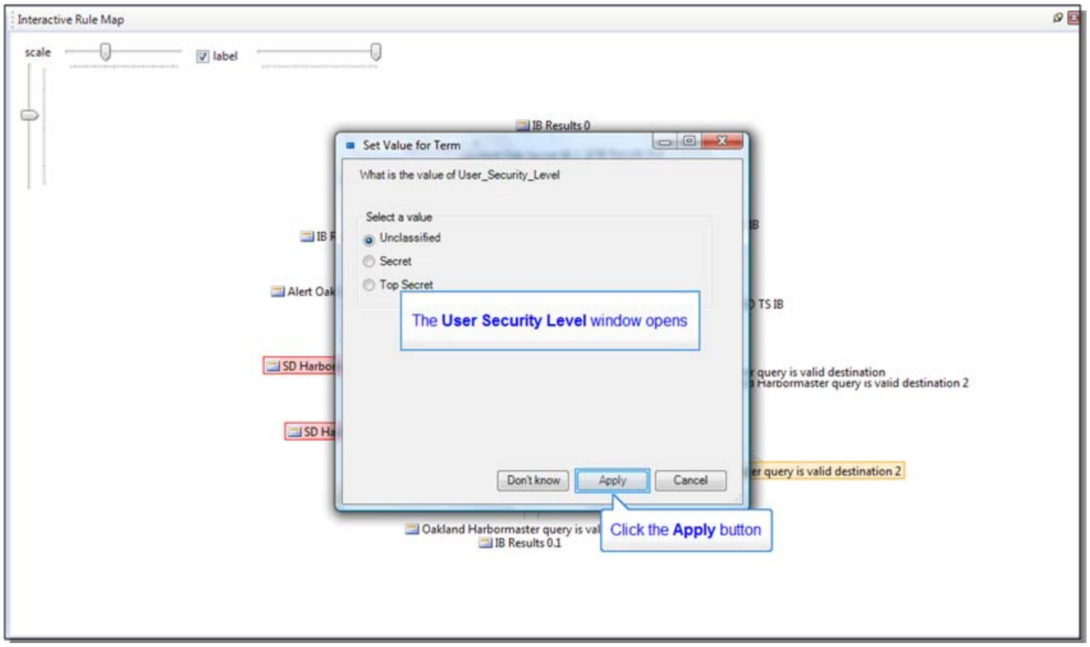
Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
IB Results 0
IB Results 4
Alert Oak Secret IB 2
IB Results 0.3
EWO query is valid
Alert SD Secret IB
IB Results 0.2
IB Results 2
Alert SD Secret IB 2
Alert OakTS IB
Alert Oak Secret IB
Alert SD TS IB 2
Alert SD TS IB
IB Results 3
User_Security_Level
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
valid 2
SD Harbormaster query is valid destination 2
IB Results 2.1
Alert OakTS IB 2
Oakland Harbormaster query is valid origination
IB Results 0.1

Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
IB Results 0
IB Results 4
Alert Oak Secret IB 2
IB Results 0.3
EWO query is valid
Alert SD Secret IB
IB Results 0.2
IB Results 2
Alert SD Secret IB 2
Alert OakTS IB
Alert Oak Secret IB
Alert SD TS IB 2
Alert SD TS IB
IB Results 3
User_Security_Level
SD Harbormaster query is valid origination
Oakland Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid origination 2
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
valid 2
SD Harbormaster query is valid destination 2
IB Results 2.1
Alert OakTS IB 2
Oakland Harbormaster query is valid origination
IB Results 0.1

Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
IB Results 0
IB R
IB
Alert Oak
TS IB
SD Harbo
SD Ha
r query is valid destination
Harbormaster query is valid destination 2
er query is valid destination 2
Oakland Harbormaster query is val
IB Results 0.1
Don't know
Apply
Cancel

Set Value for Term

What is the value of User_Security_Level

Select a value

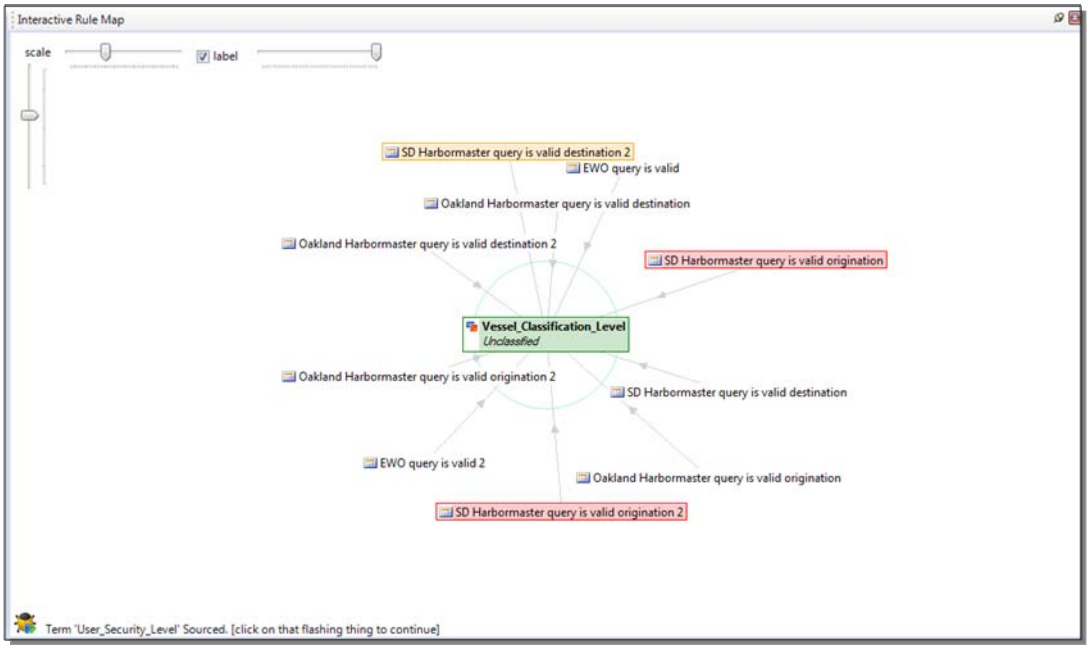
● Unclassified

○ Secret

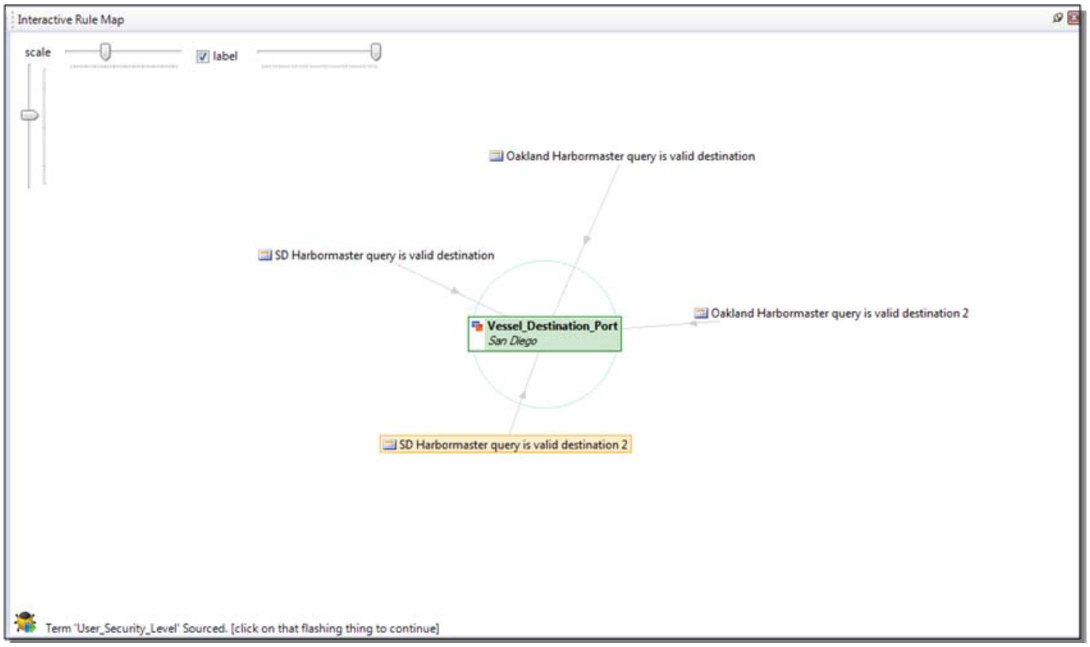
○ Top Secret

The **User Security Level** window opens

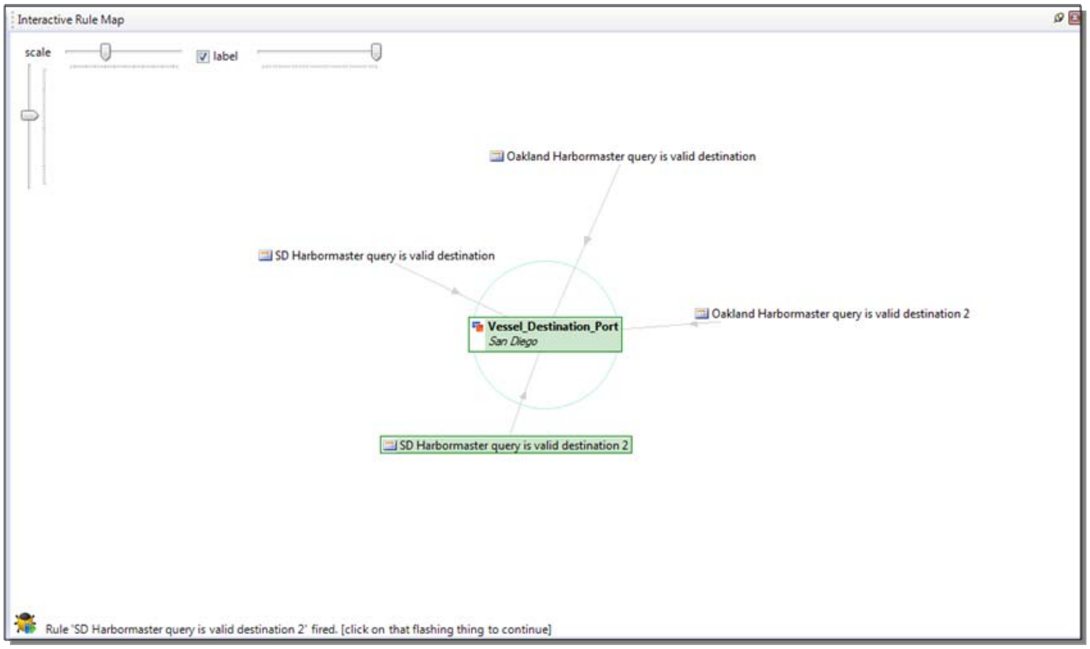
Click the **Apply** button


Interactive Rule Map
scale
☑ label
SD Harbormaster query is valid destination 2
EWO query is valid
Oakland Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid origination
Vessel_Classification_Level
Unclassified
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid destination
EWO query is valid 2
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid origination 2

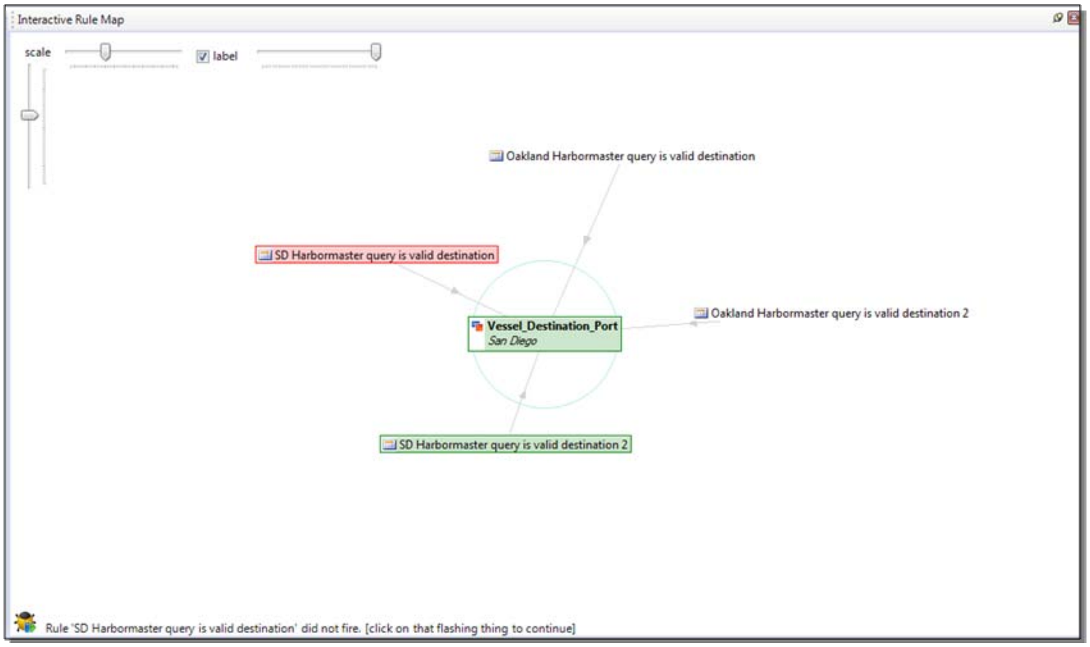
Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

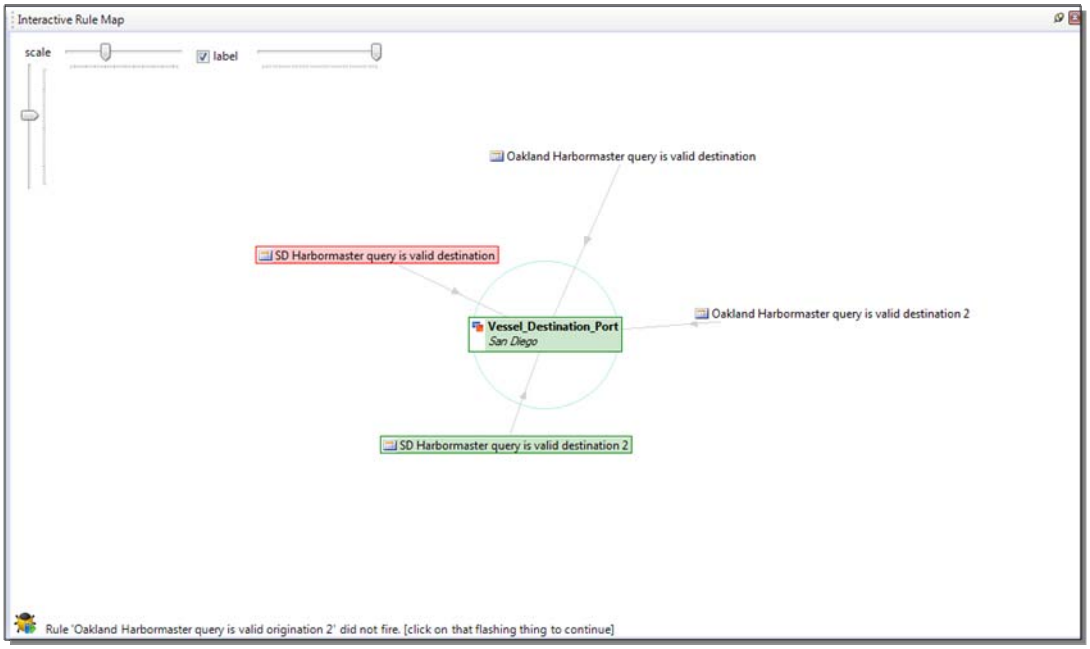
Term 'User_Security_Level' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

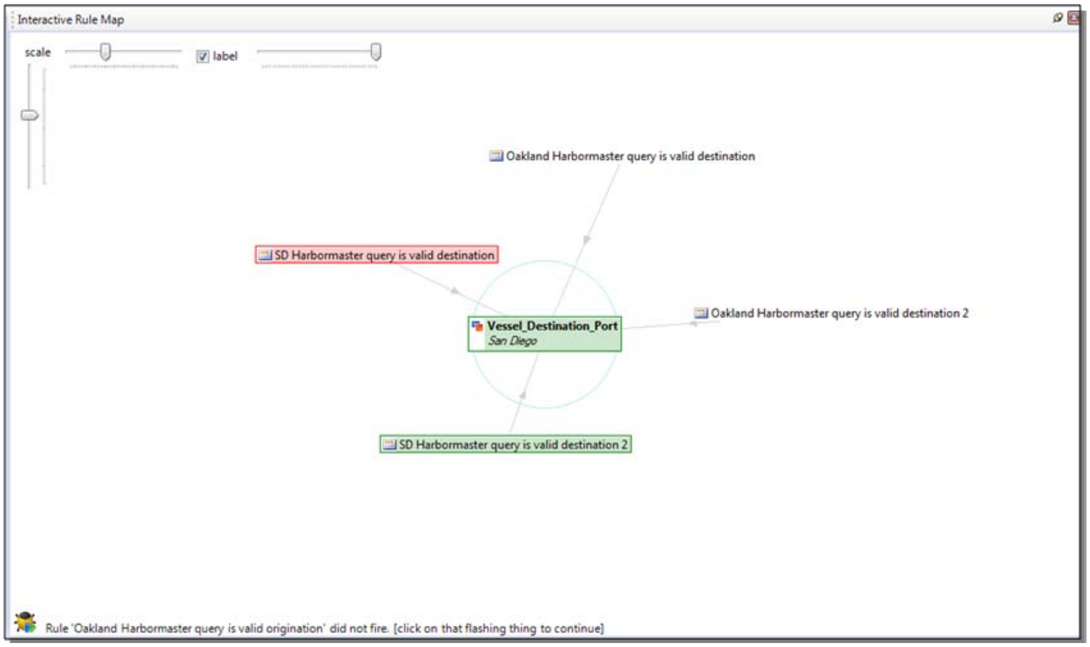
Rule 'SD Harbormaster query is valid destination 2' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

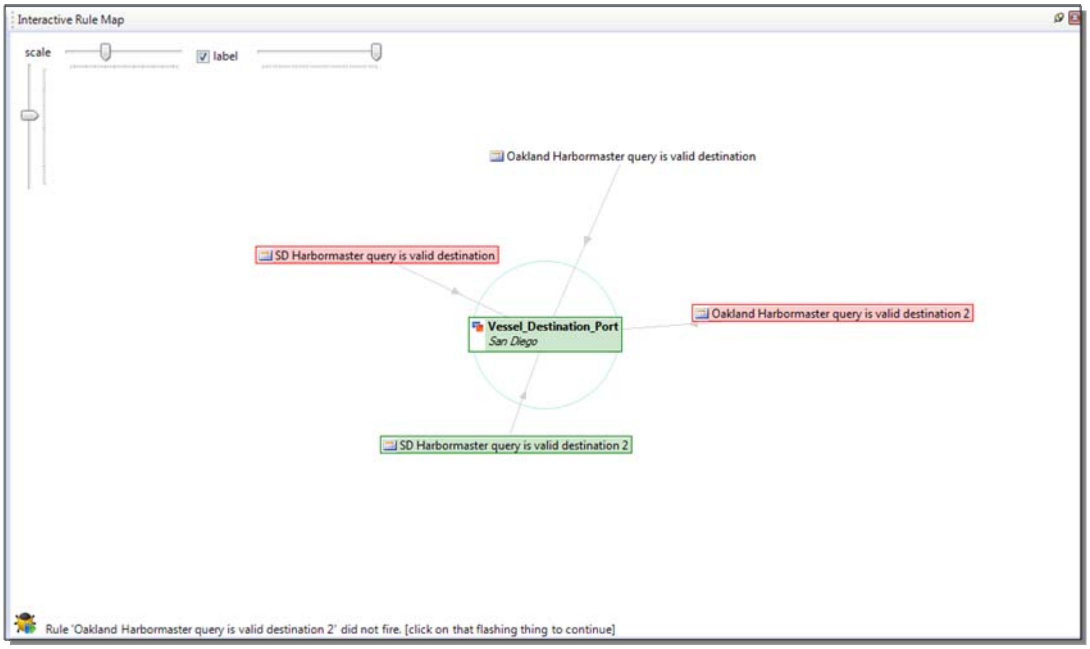
Rule 'SD Harbormaster query is valid destination' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

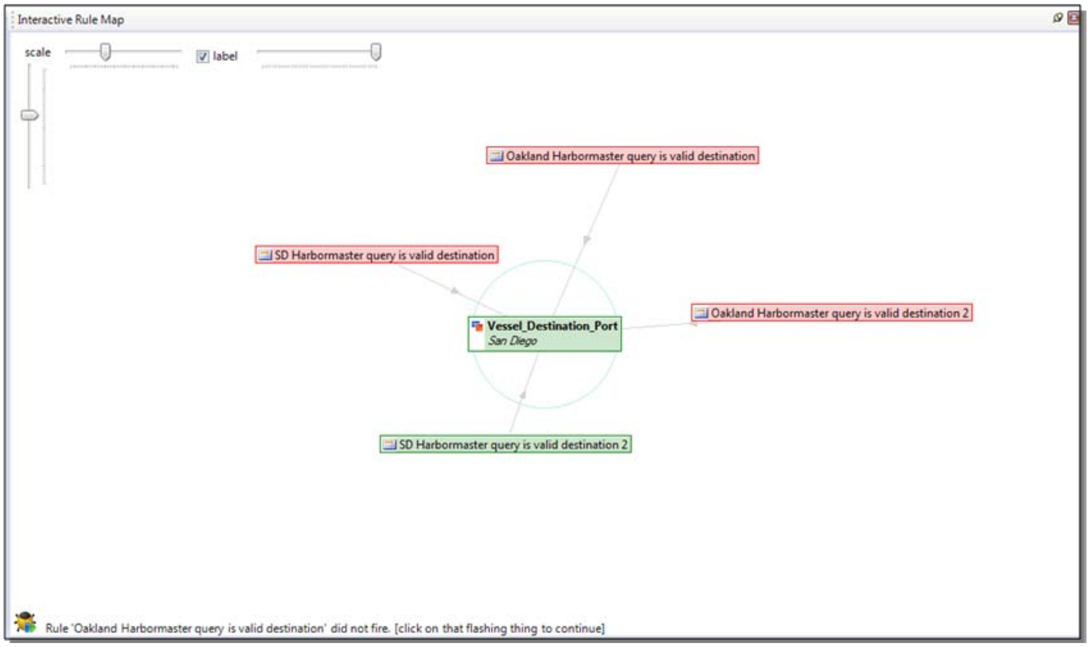
Rule 'Oakland Harbormaster query is valid origination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
SD Harbormaster query is valid destination 2

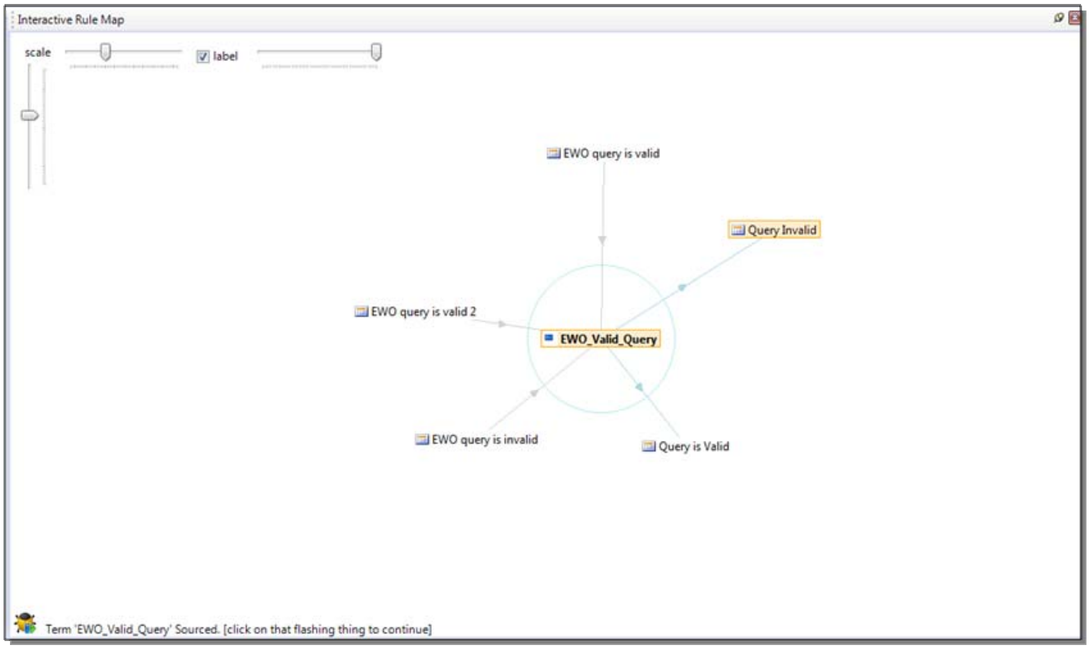
Rule 'Oakland Harbormaster query is valid origination' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
Vessel_Destination_Port
San Diego
SD Harbormaster query is valid destination 2

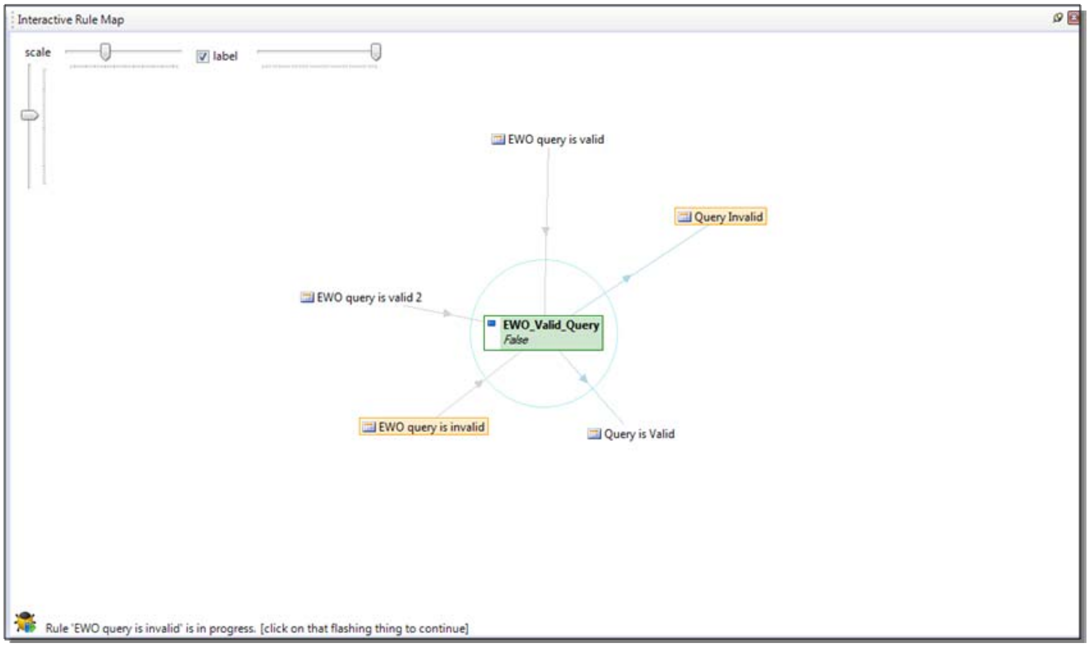
Rule 'Oakland Harbormaster query is valid destination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
SD Harbormaster query is valid destination
Vessel_Destination_Port
San Diego
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination 2

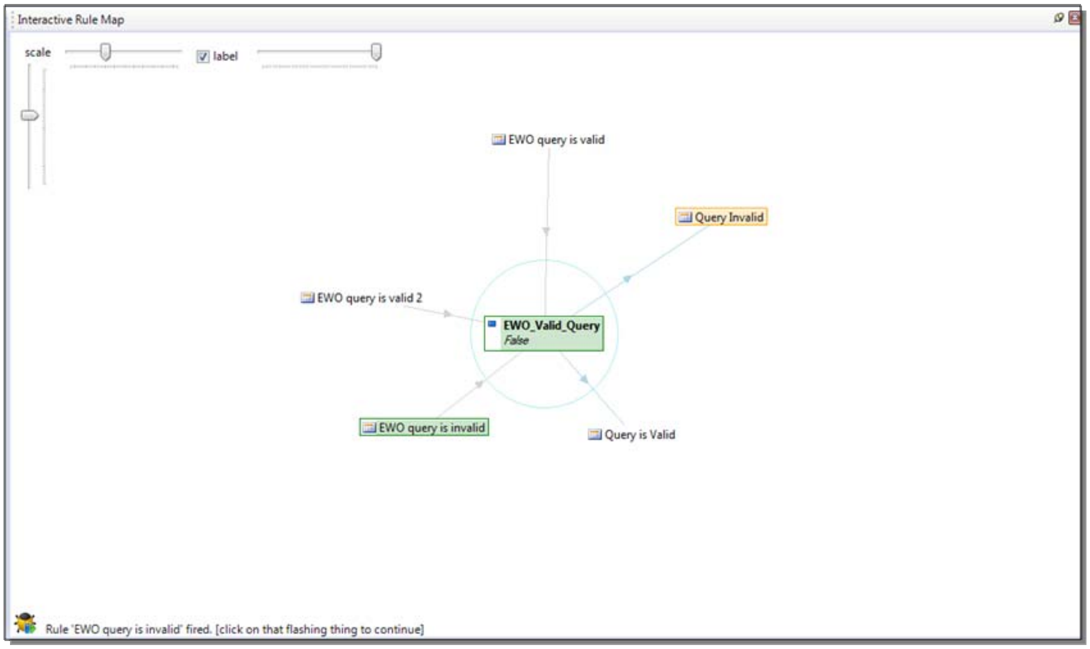
Rule 'Oakland Harbormaster query is valid destination' did not fire. [click on that flashing thing to continue]



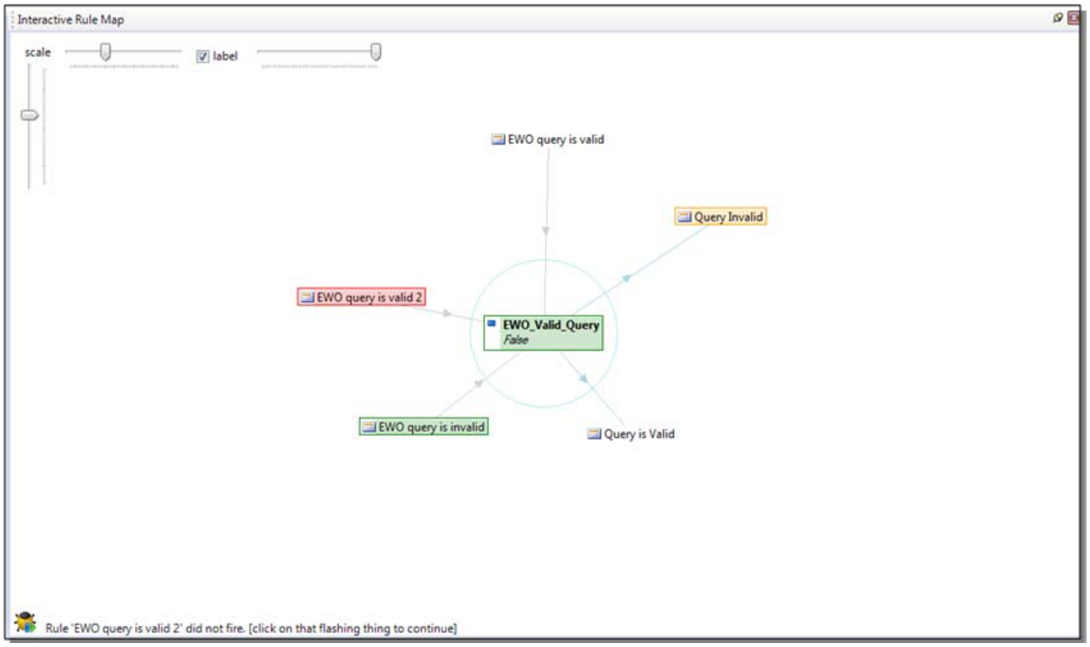
Term 'EWO_Valid_Query' Sourced. [click on that flashing thing to continue]



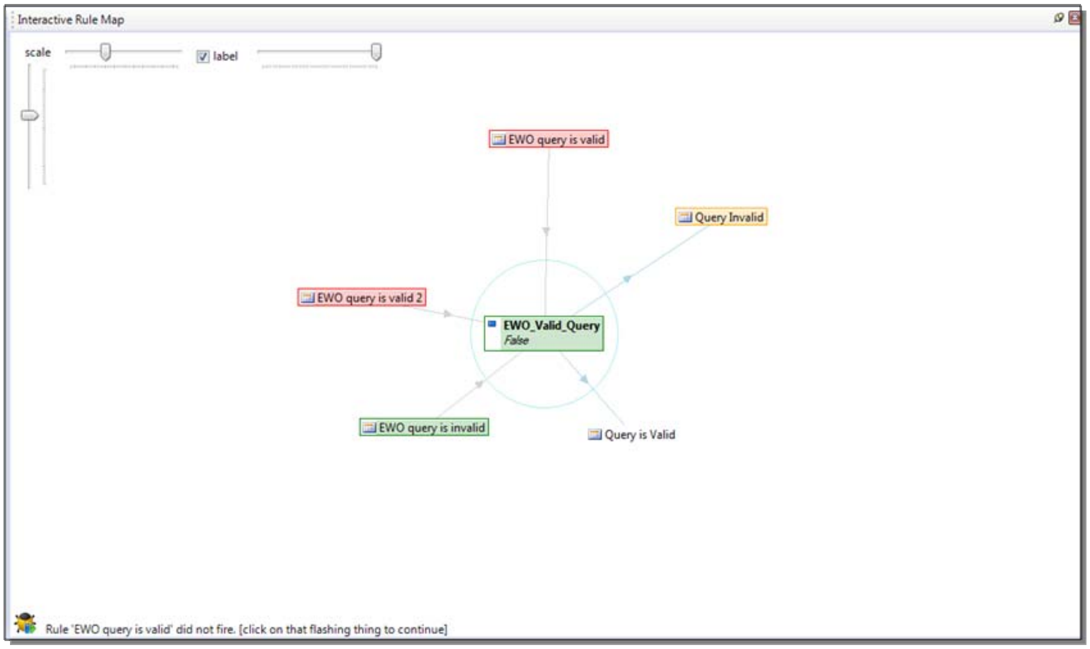
Rule 'EWO query is invalid' is in progress. [click on that flashing thing to continue]



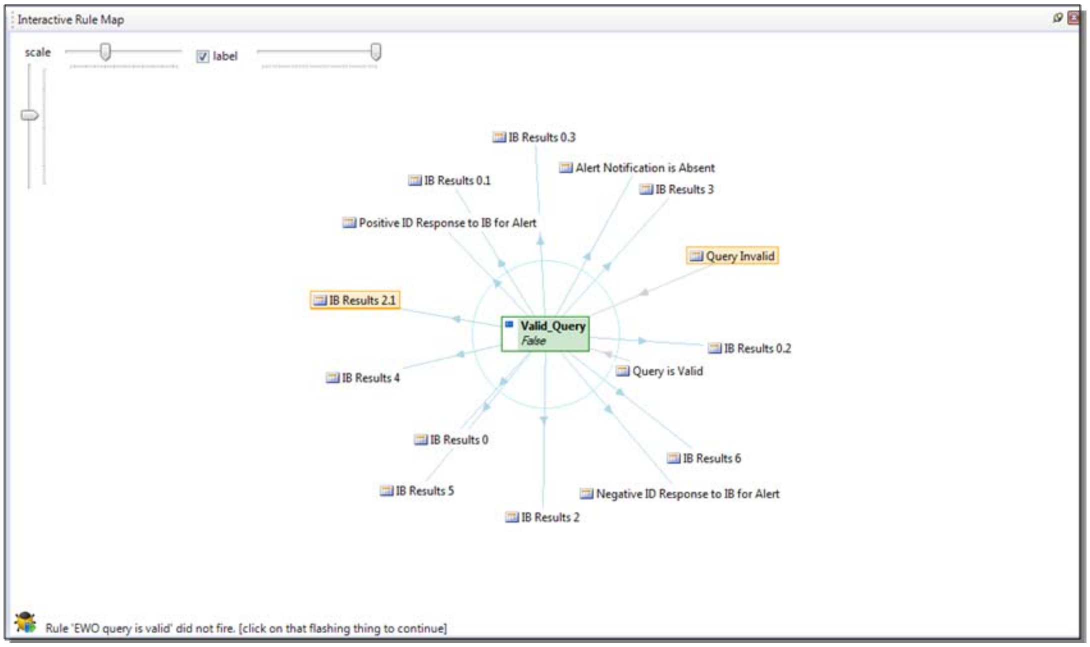
Rule 'EWO query is invalid' fired. [click on that flashing thing to continue]



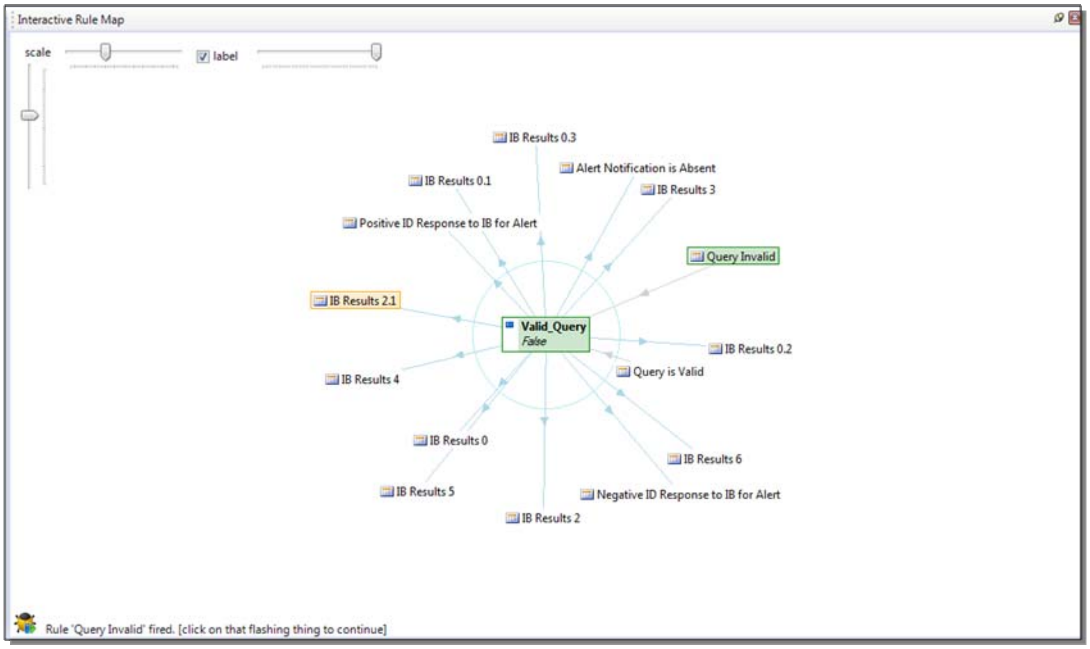
Rule 'EWO query is valid 2' did not fire. [click on that flashing thing to continue]



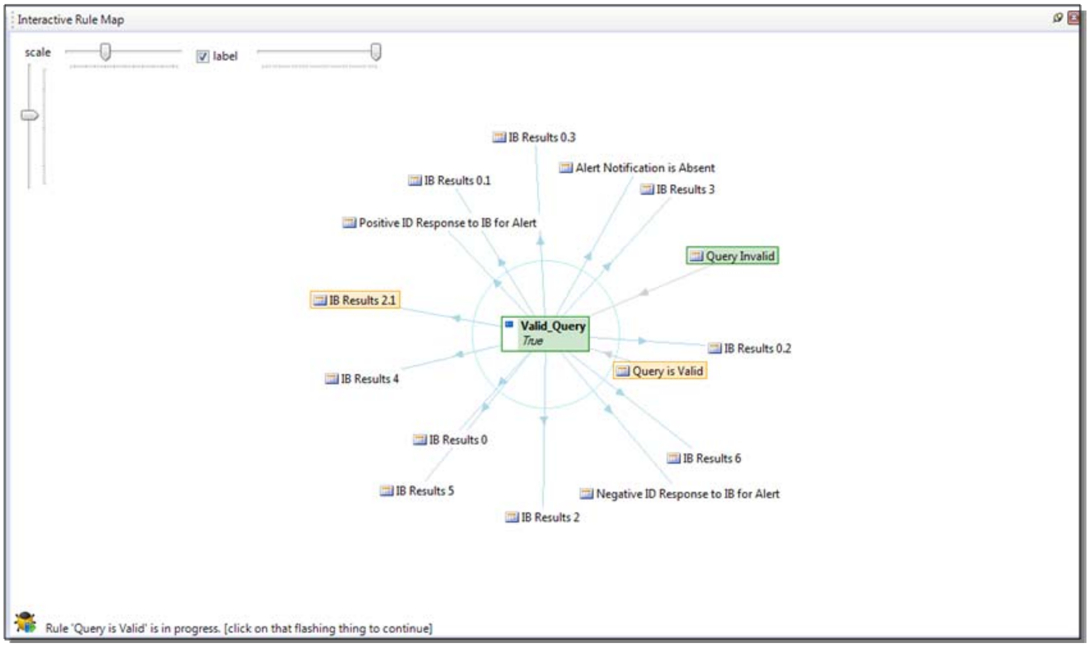
Rule 'EWO query is valid' did not fire. [click on that flashing thing to continue]



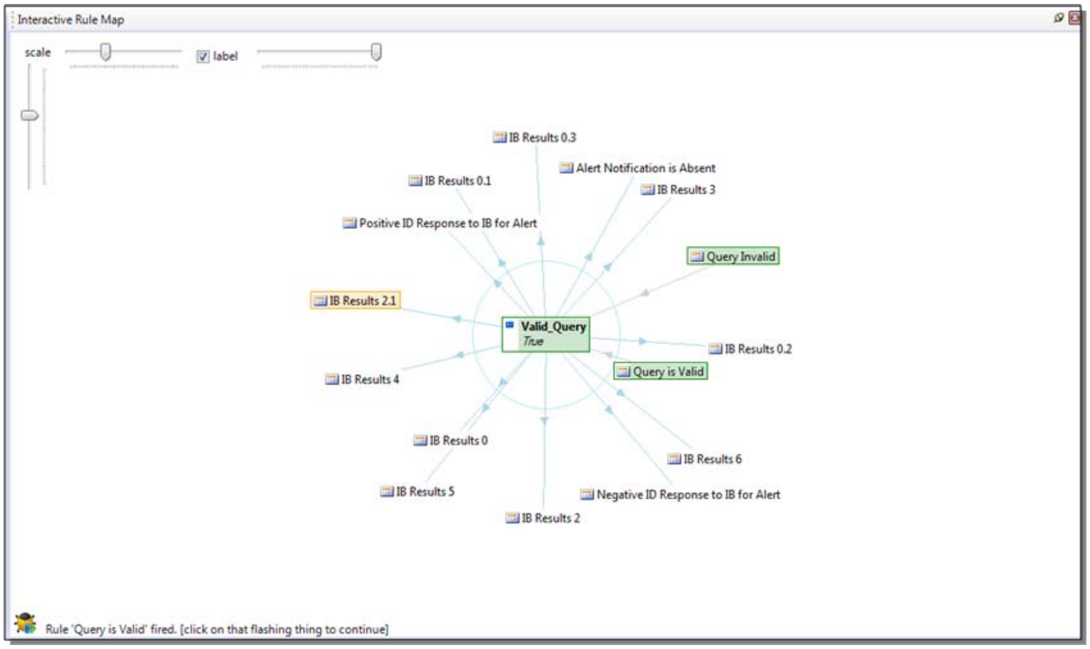
Rule 'EWO query is valid' did not fire. [click on that flashing thing to continue]



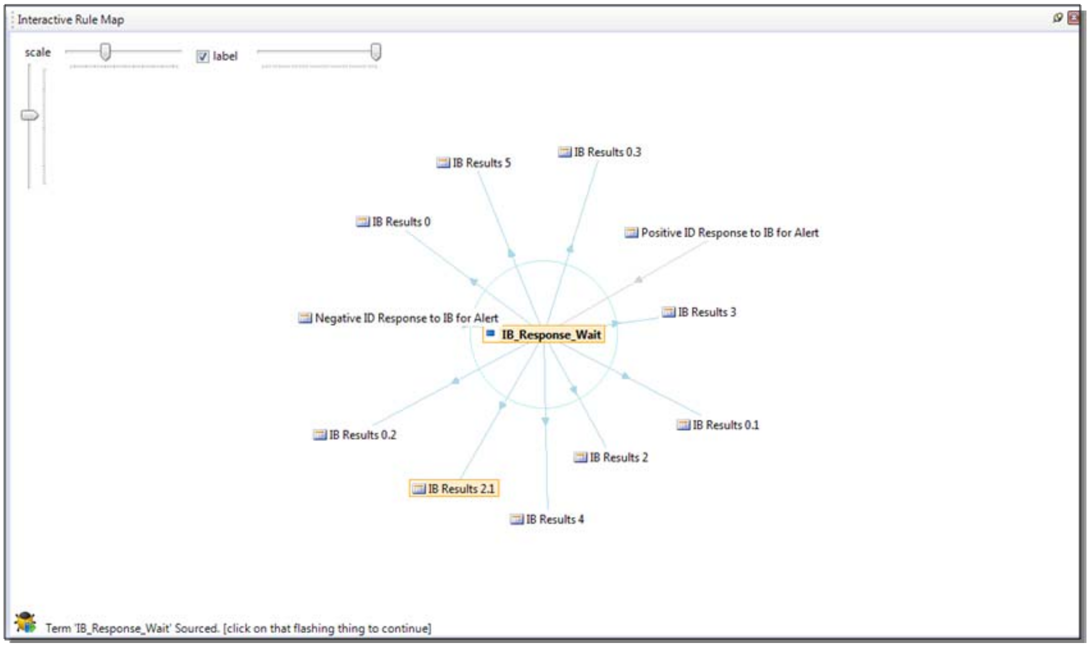
Rule 'Query Invalid' fired. [click on that flashing thing to continue]



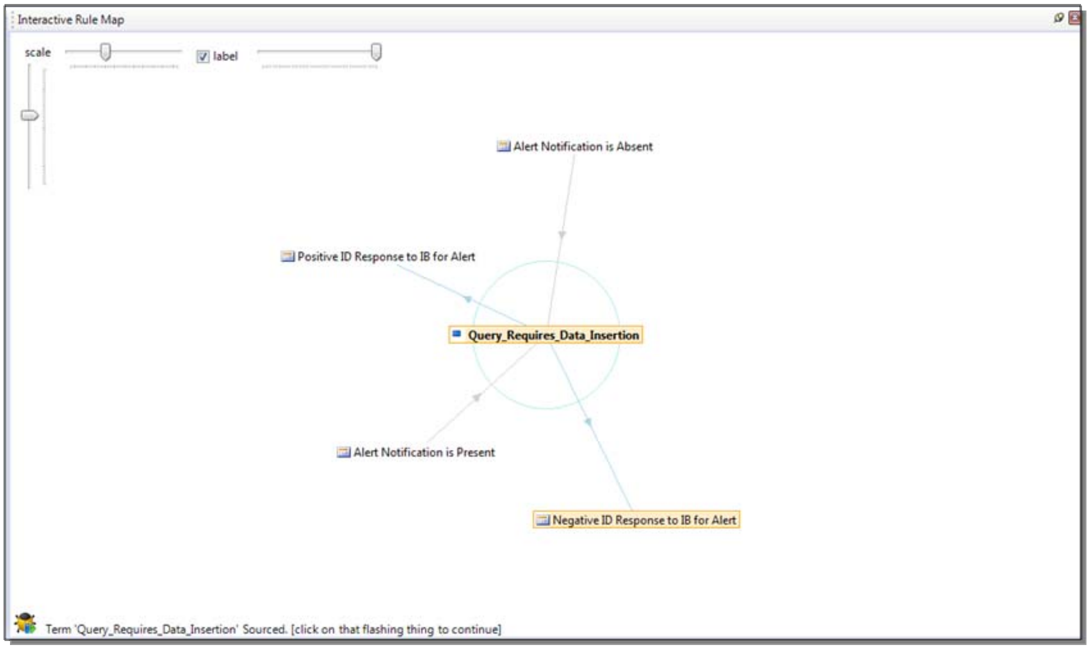
Rule 'Query is Valid' is in progress. [click on that flashing thing to continue]



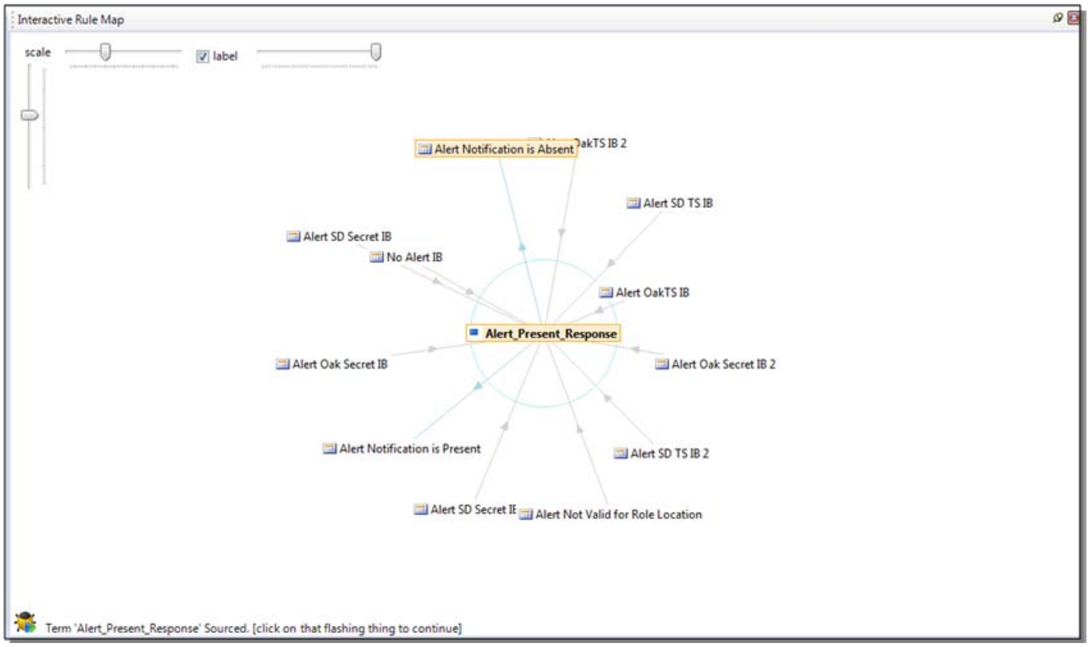
Rule 'Query is Valid' fired. [click on that flashing thing to continue]



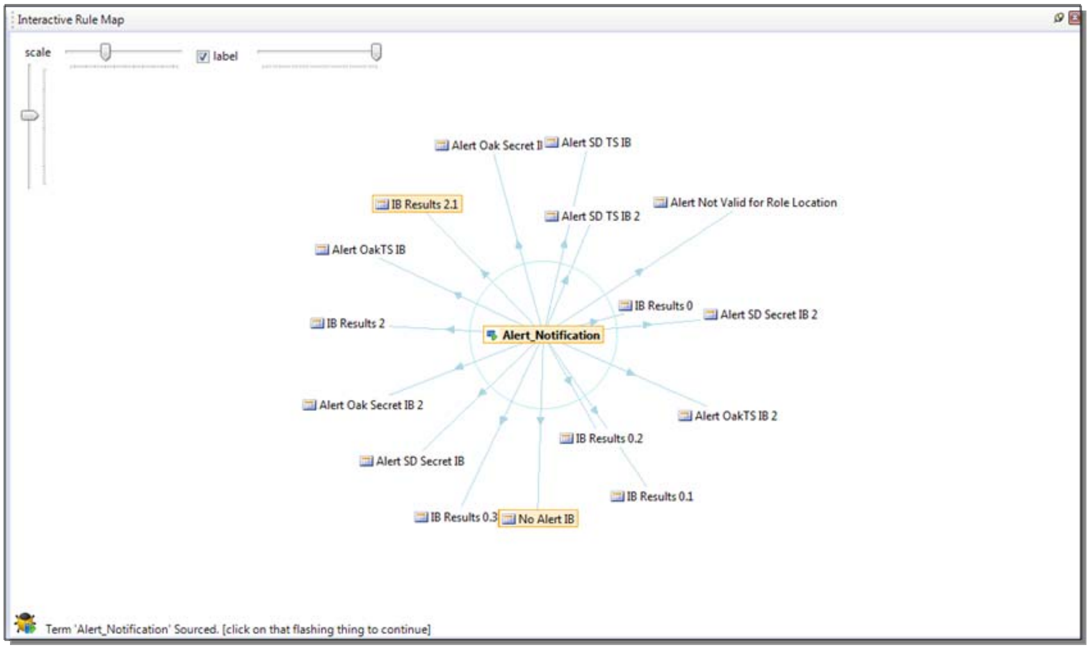
Term 'IB_Response_Wait' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Alert Notification is Absent
Positive ID Response to IB for Alert
Query_Requires_Data_Insertion
Alert Notification is Present
Negative ID Response to IB for Alert

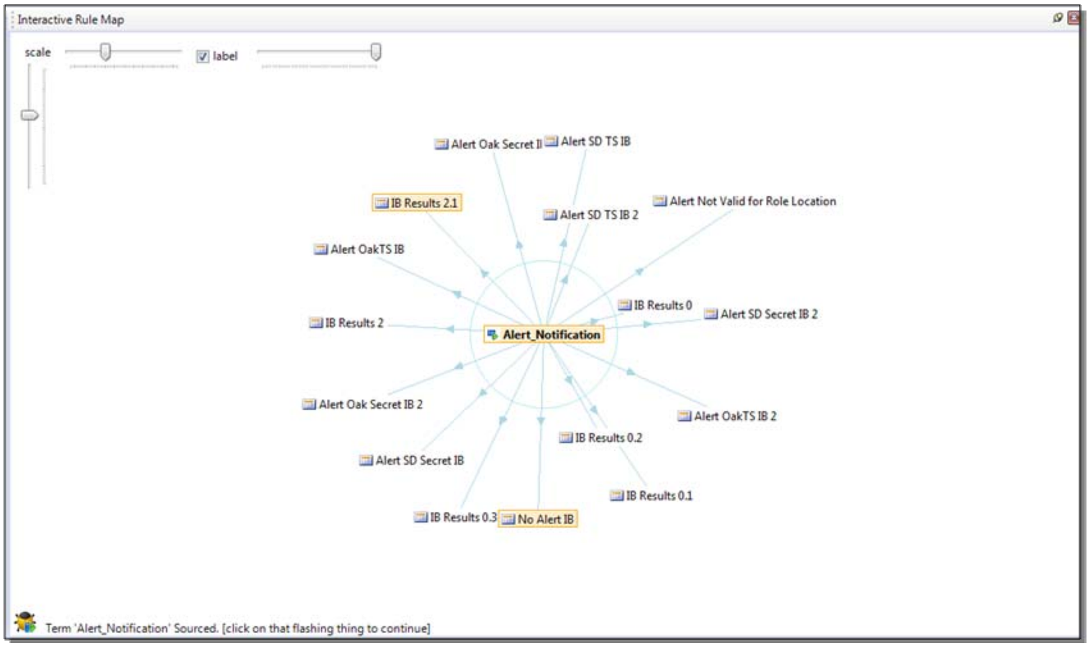
Term 'Query_Requires_Data_Insertion' Sourced. [click on that flashing thing to continue]



Term 'Alert_Present_Response' Sourced. [click on that flashing thing to continue]



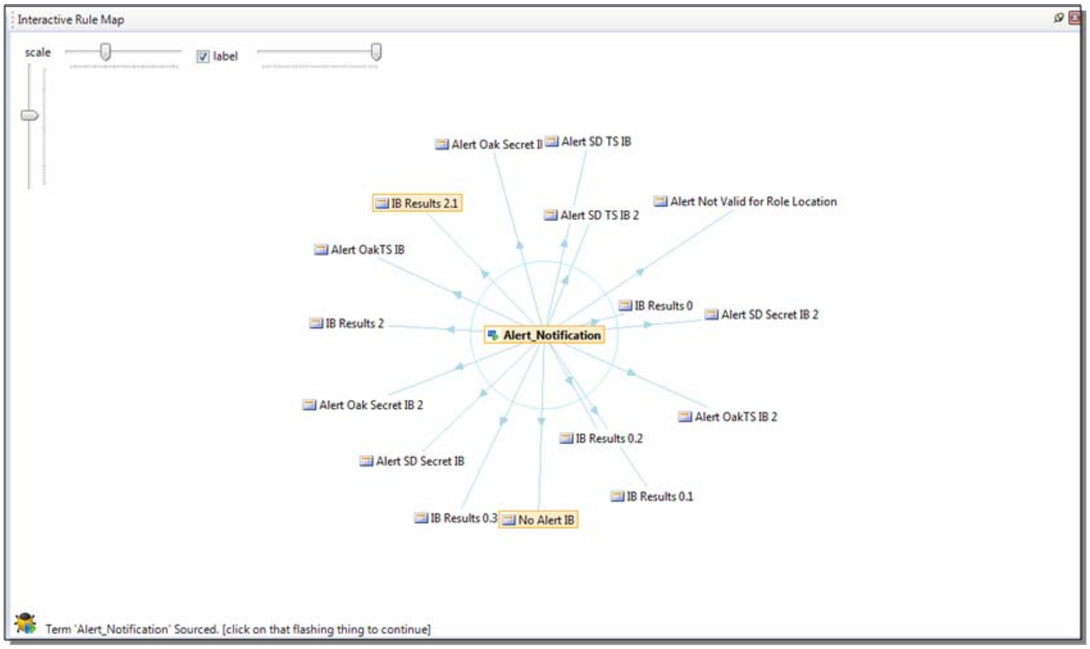
Term 'Alert_Notification' Sourced. [click on that flashing thing to continue]



Interactive Rule Map

scale ☑ label

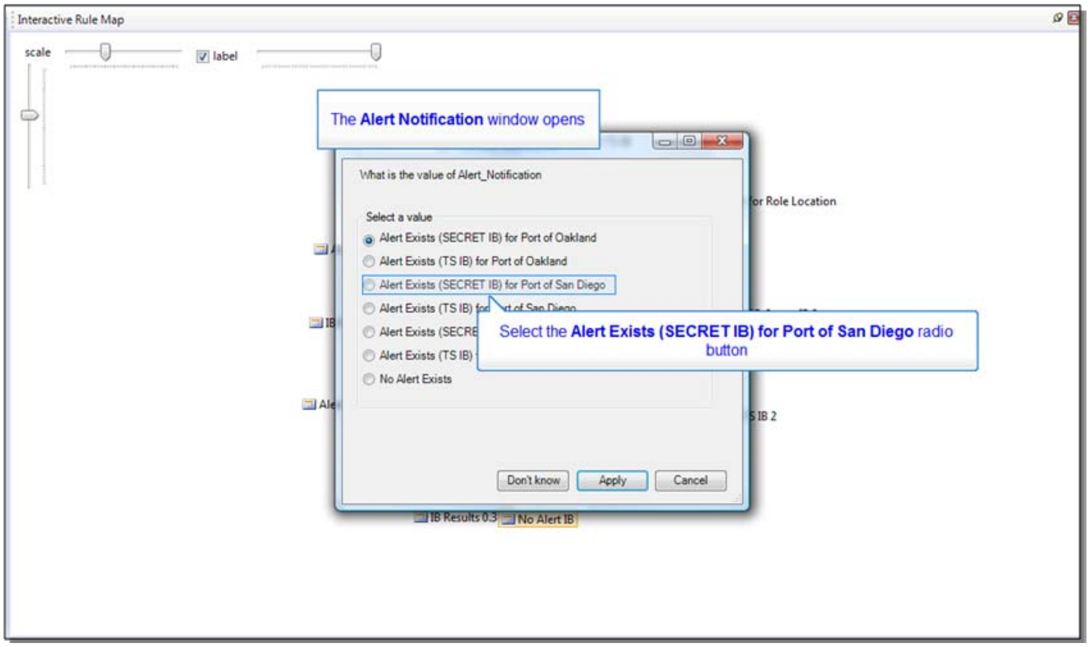
Term 'Alert_Notification' Sourced. [click on that flashing thing to continue]



Interactive Rule Map

scale ☑ label

Term 'Alert_Notification' Sourced. [click on that flashing thing to continue]



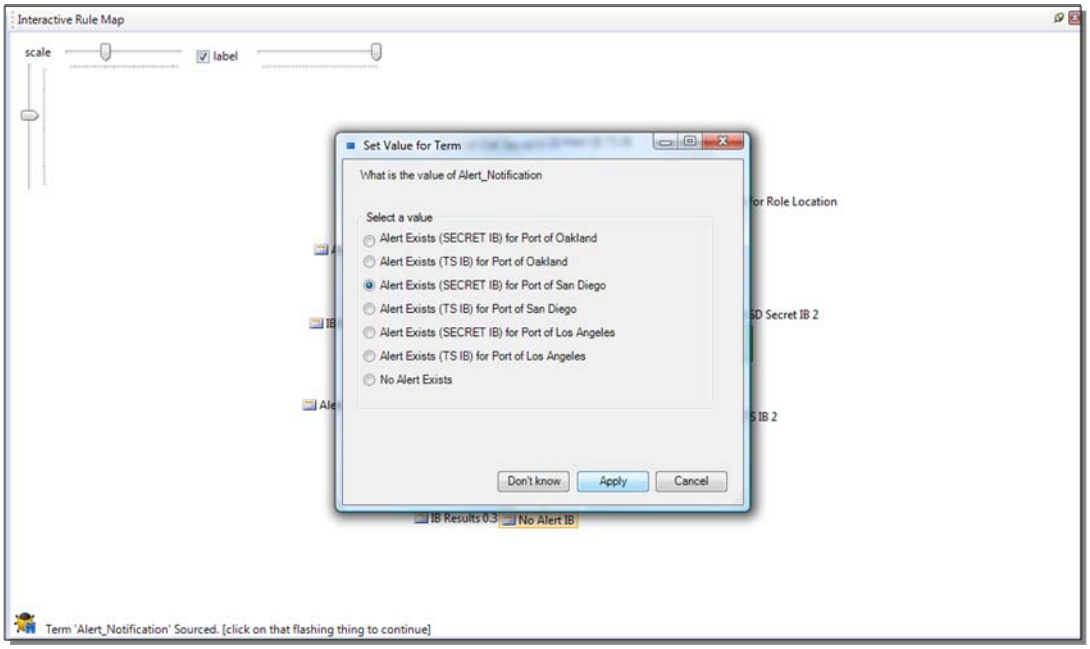
The **Alert Notification** window opens

What is the value of Alert_Notification

Select a value

● Alert Exists (SECRET IB) for Port of Oakland
○ Alert Exists (TS IB) for Port of Oakland
○ Alert Exists (SECRET IB) for Port of San Diego
○ Alert Exists (TS IB) for
○ Alert Exists (SECRE
○ Alert Exists (TS IB)
○ No Alert Exists

Select the **Alert Exists (SECRET IB) for Port of San Diego** radio button



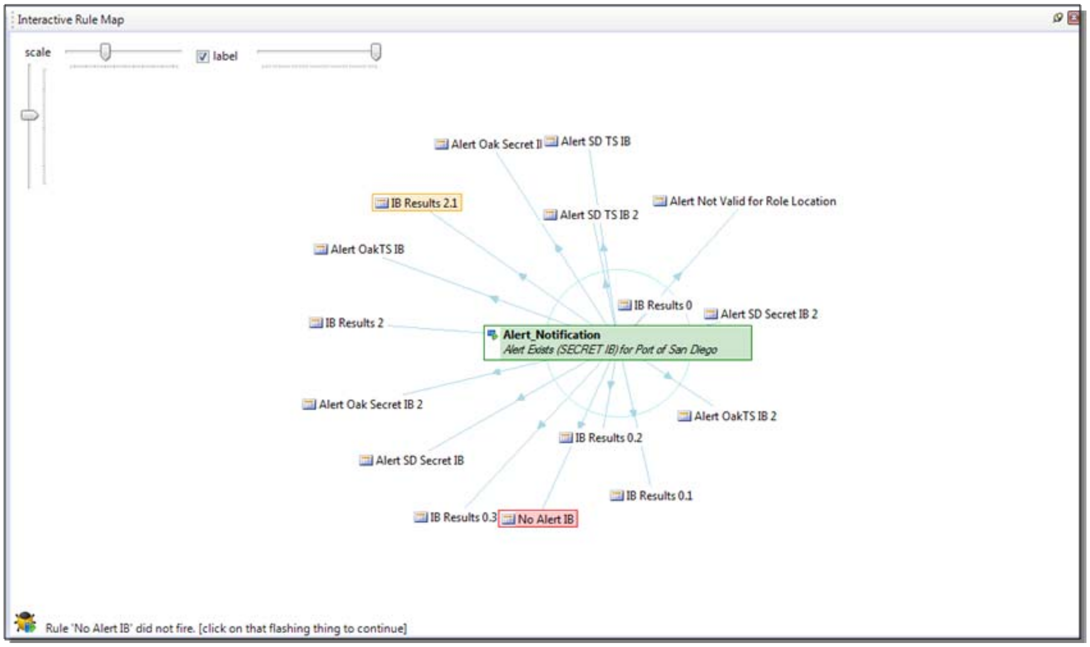
Set Value for Term

What is the value of Alert_Notification

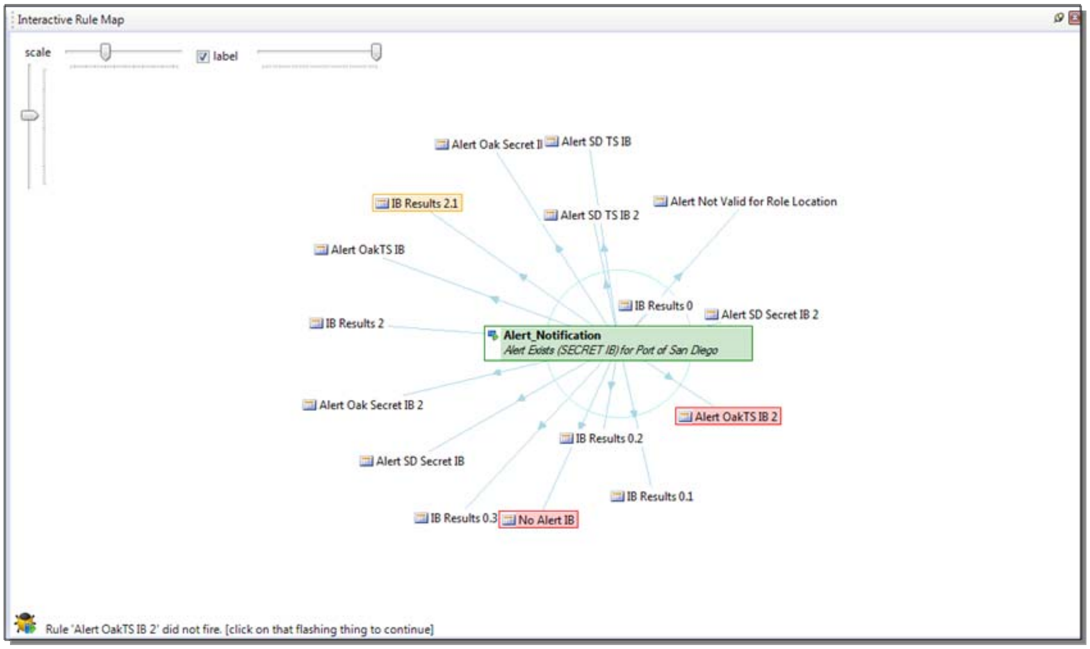
Select a value

○ Alert Exists (SECRET IB) for Port of Oakland
○ Alert Exists (TS IB) for Port of Oakland
● Alert Exists (SECRET IB) for Port of San Diego
○ Alert Exists (TS IB) for Port of San Diego
○ Alert Exists (SECRET IB) for Port of Los Angeles
○ Alert Exists (TS IB) for Port of Los Angeles
○ No Alert Exists

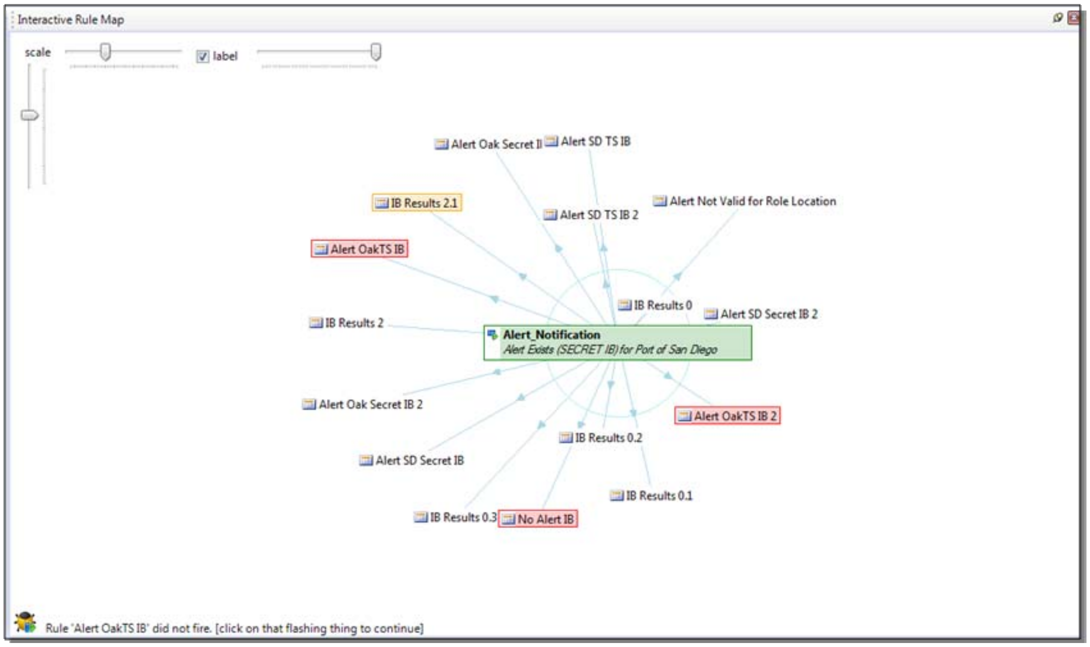
Term 'Alert_Notification' Sourced. [click on that flashing thing to continue]



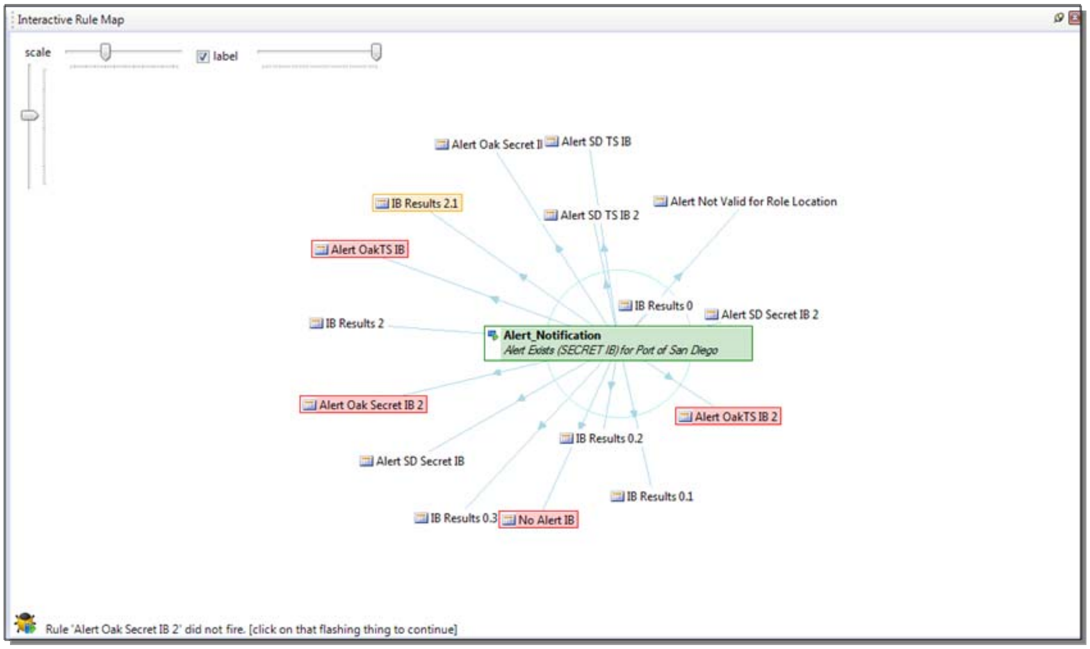
Rule 'No Alert IB' did not fire. [click on that flashing thing to continue]



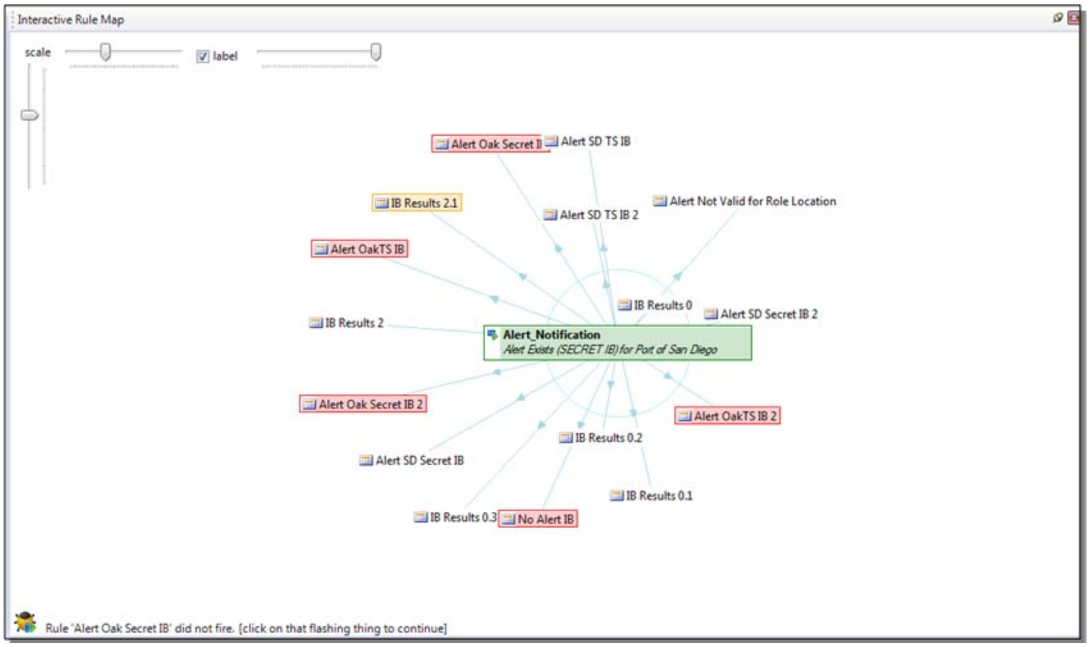
Rule 'Alert OakTS IB 2' did not fire. [click on that flashing thing to continue]



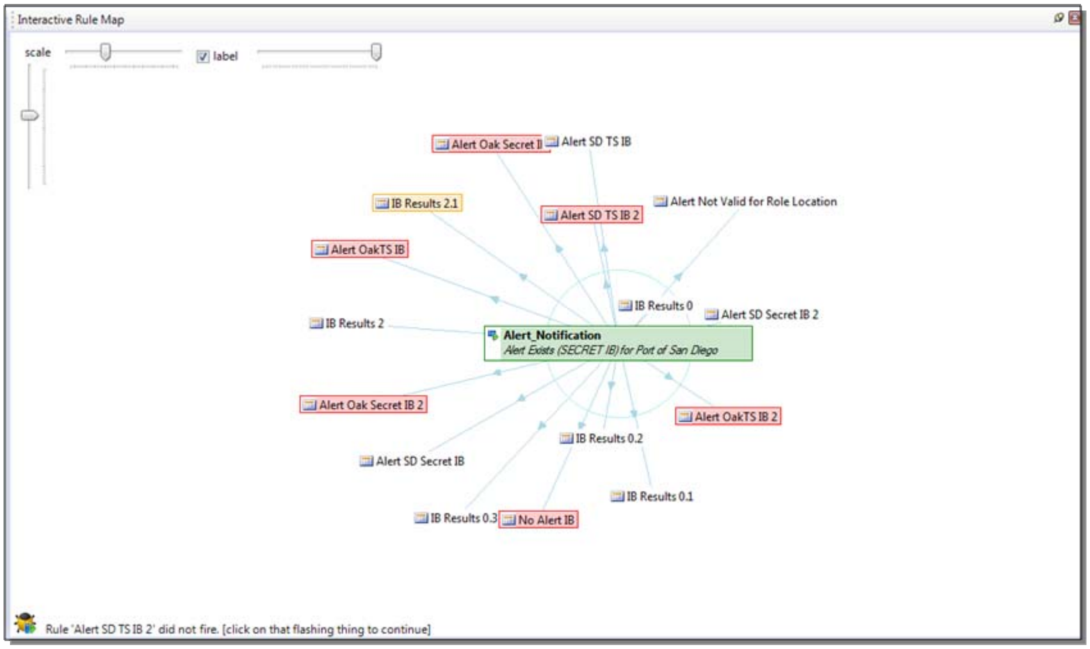
Rule 'Alert OakTS IB' did not fire. [click on that flashing thing to continue]



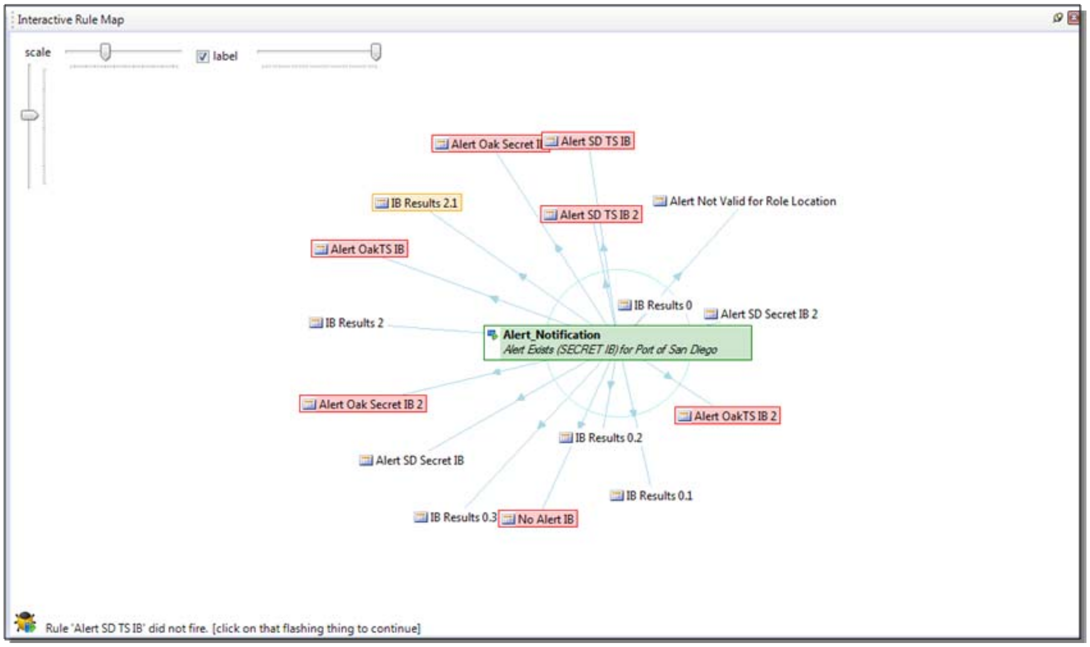
Rule 'Alert Oak Secret IB 2' did not fire. [click on that flashing thing to continue]



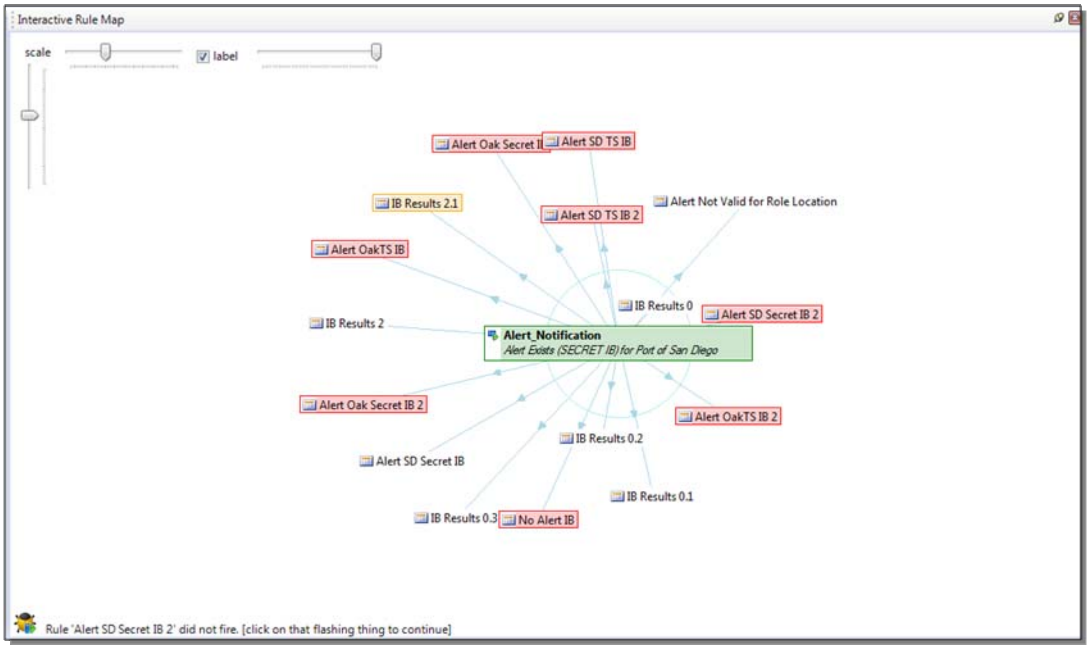
Rule 'Alert Oak Secret IB' did not fire. [click on that flashing thing to continue]



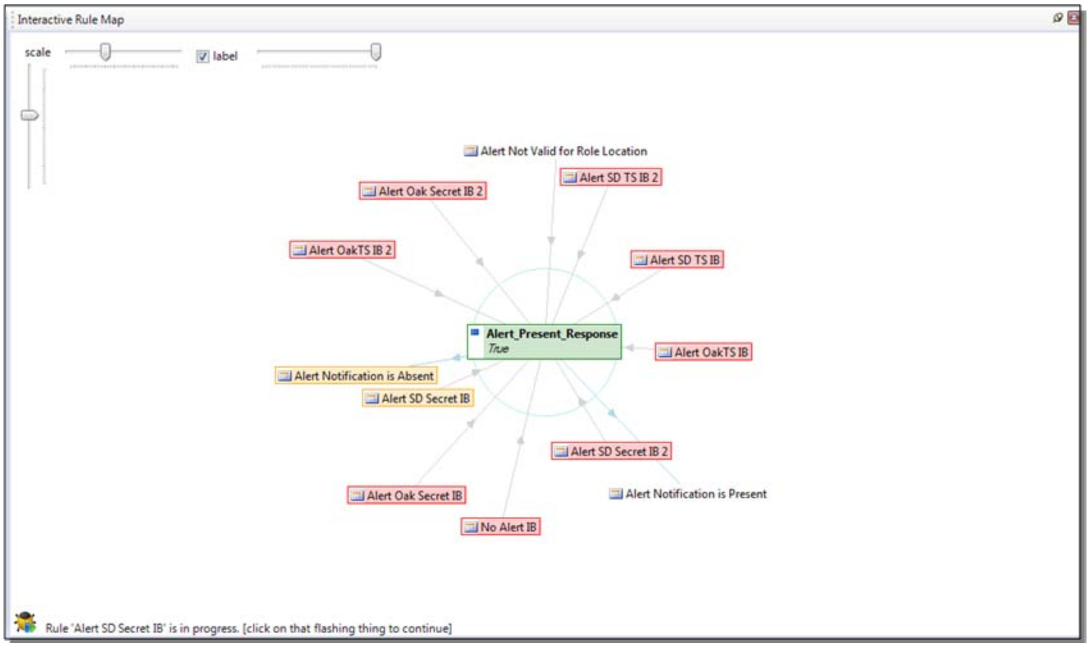
Rule 'Alert SD TS IB 2' did not fire. [click on that flashing thing to continue]



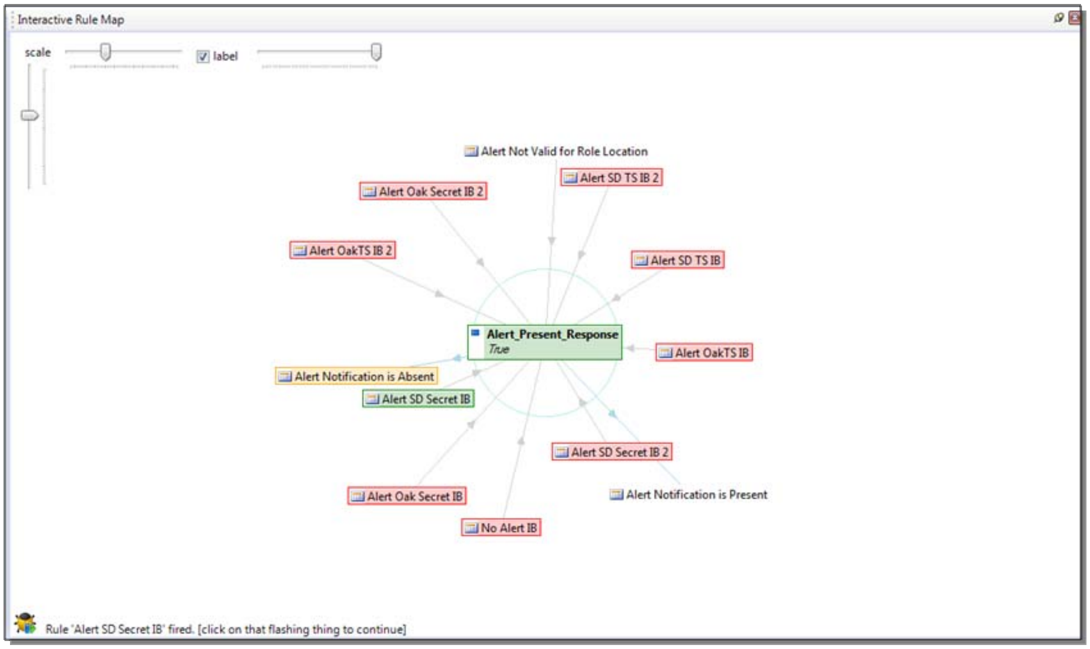
Rule 'Alert SD TS IB' did not fire. [click on that flashing thing to continue]



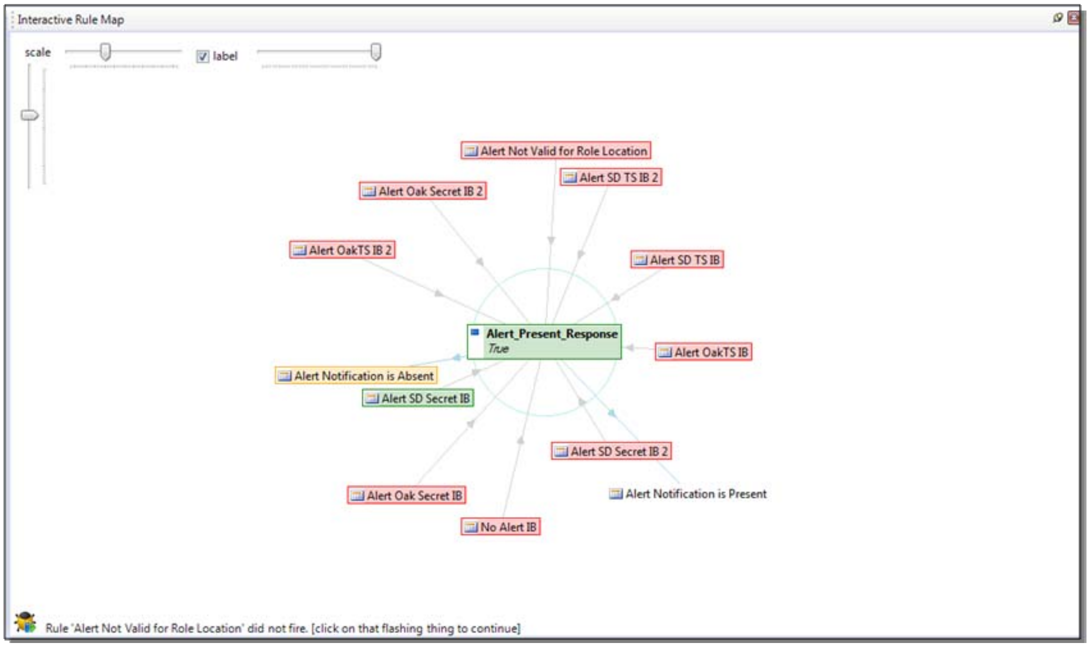
Rule 'Alert SD Secret IB 2' did not fire. [click on that flashing thing to continue]



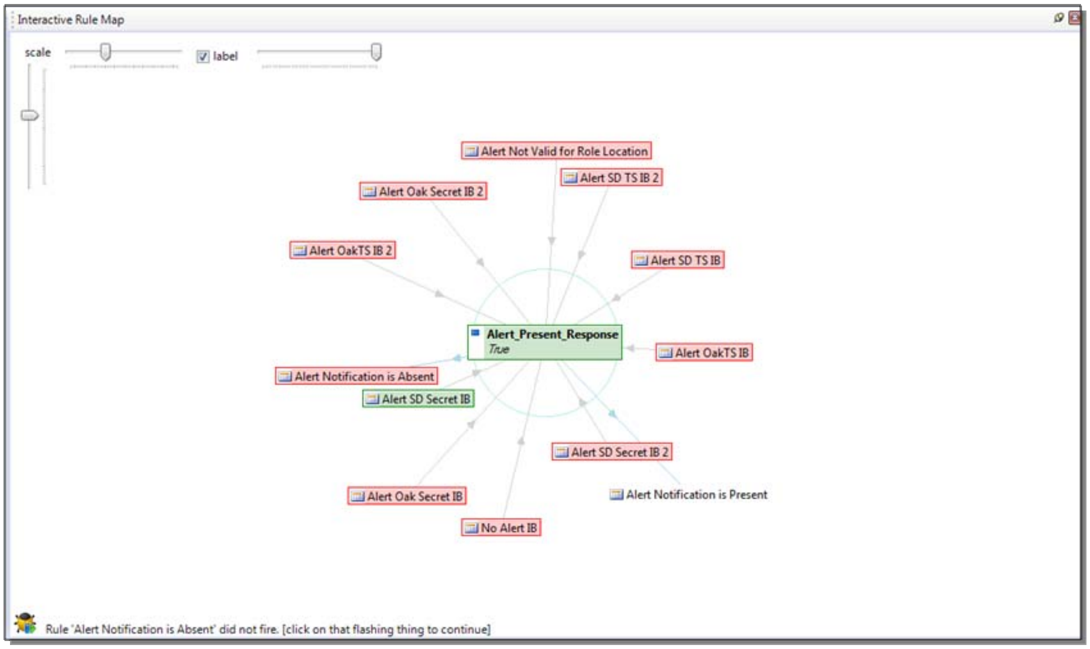
Rule 'Alert SD Secret IB' is in progress. [click on that flashing thing to continue]



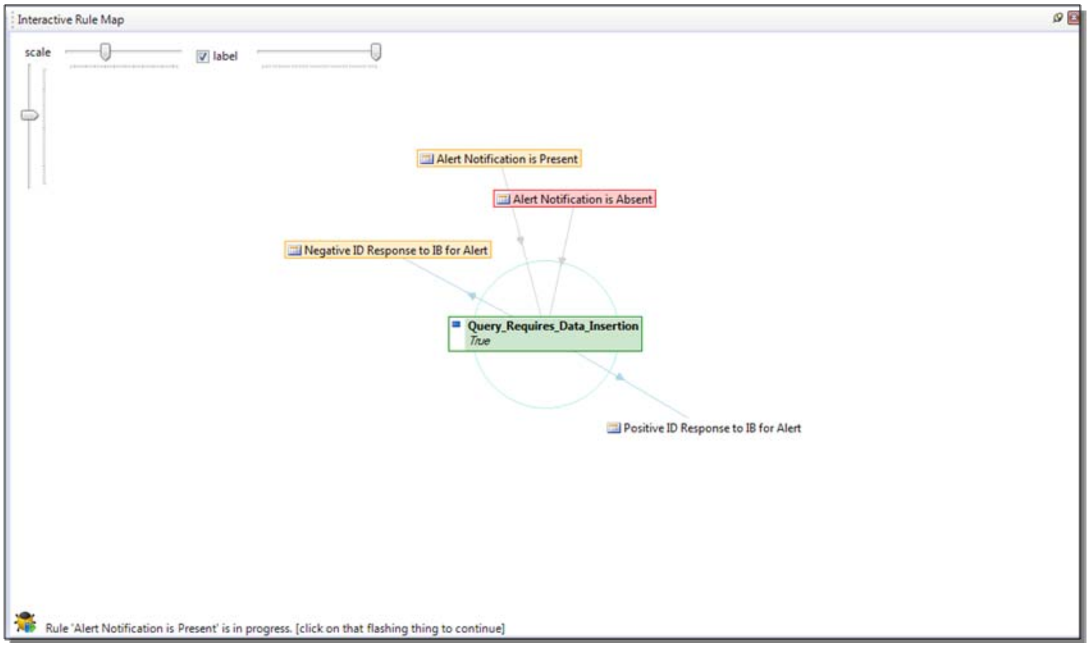
Rule 'Alert SD Secret IB' fired. [click on that flashing thing to continue]



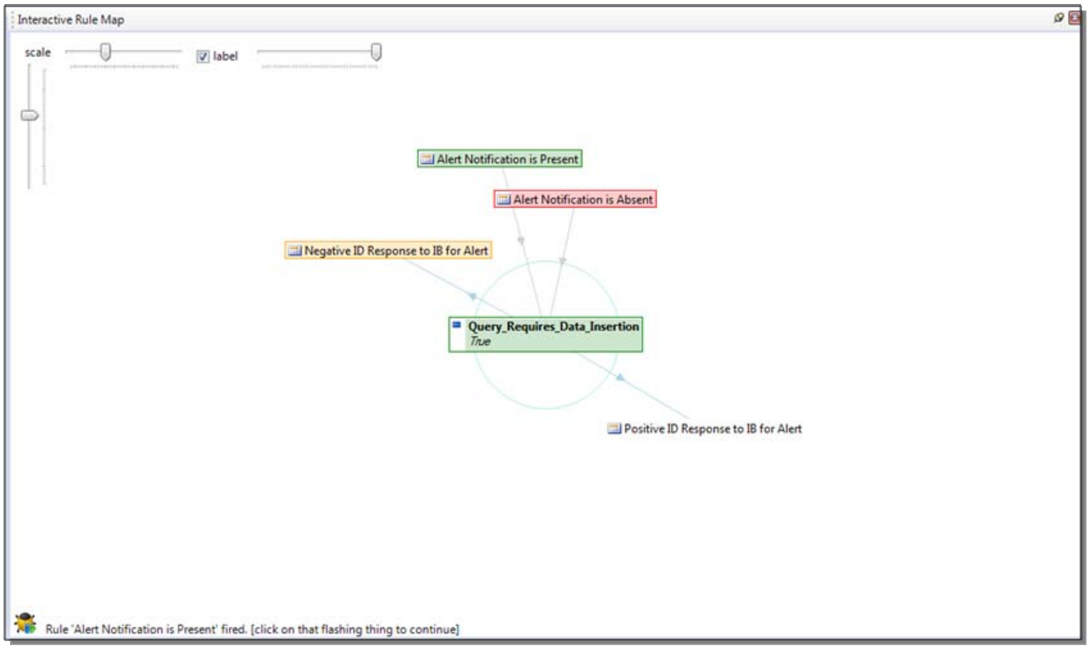
Rule 'Alert Not Valid for Role Location' did not fire. [click on that flashing thing to continue]



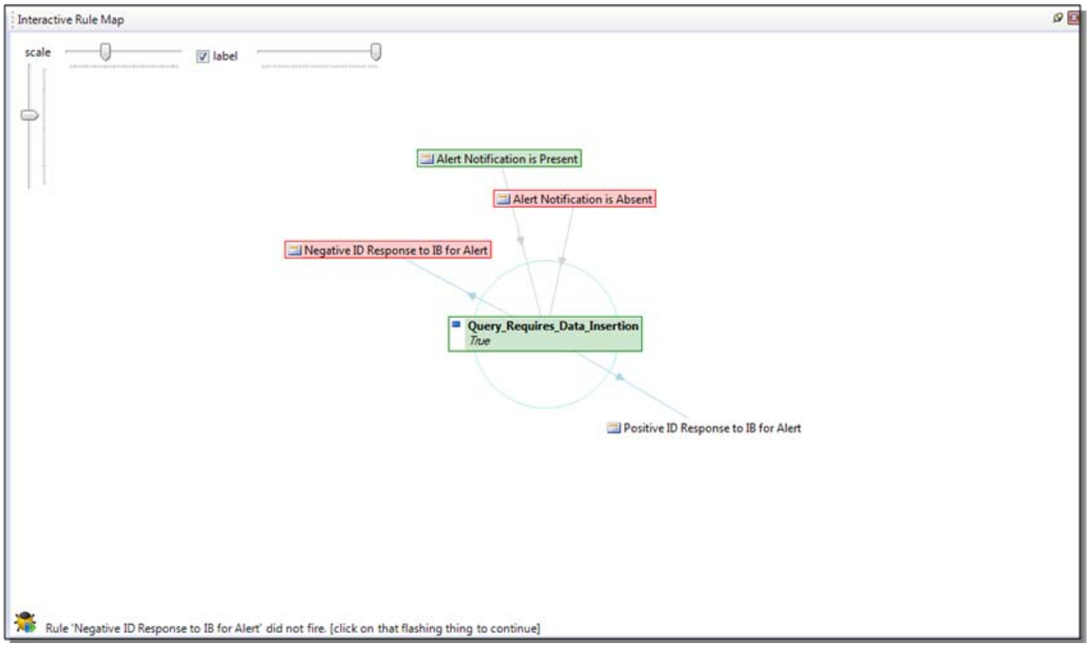
Rule 'Alert Notification is Absent' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
label
Alert Notification is Present
Alert Notification is Absent
Negative ID Response to IB for Alert
Query_Requires_Data_Insertion
True
Positive ID Response to IB for Alert

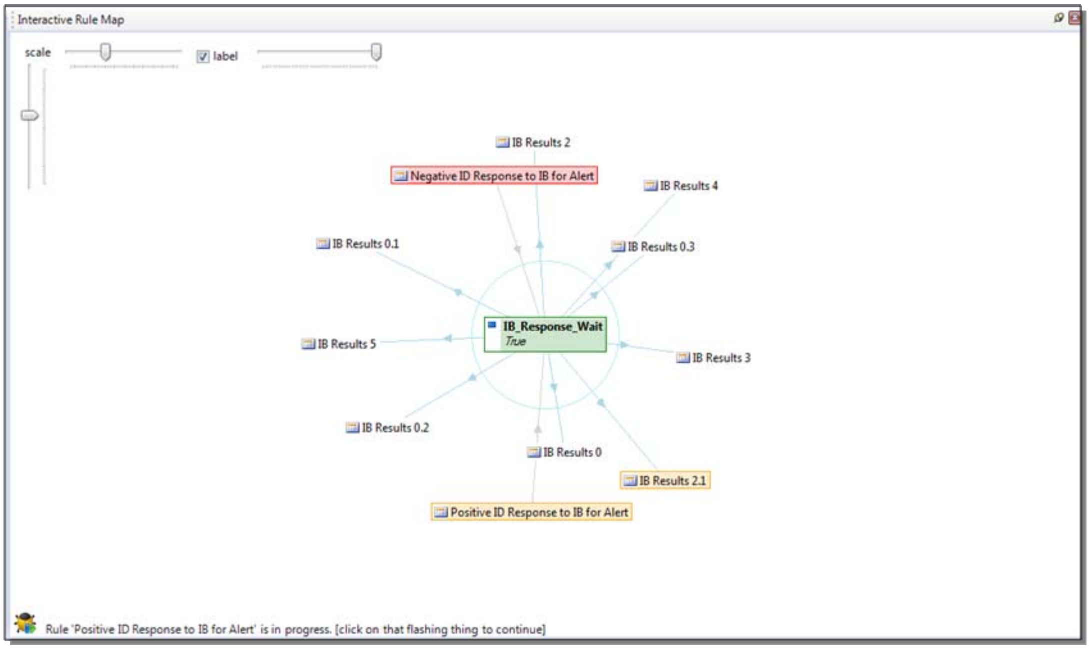
Rule 'Alert Notification is Present' is in progress. [click on that flashing thing to continue]


Interactive Rule Map
scale
label
Alert Notification is Present
Alert Notification is Absent
Negative ID Response to IB for Alert
Query_Requires_Data_Insertion
True
Positive ID Response to IB for Alert

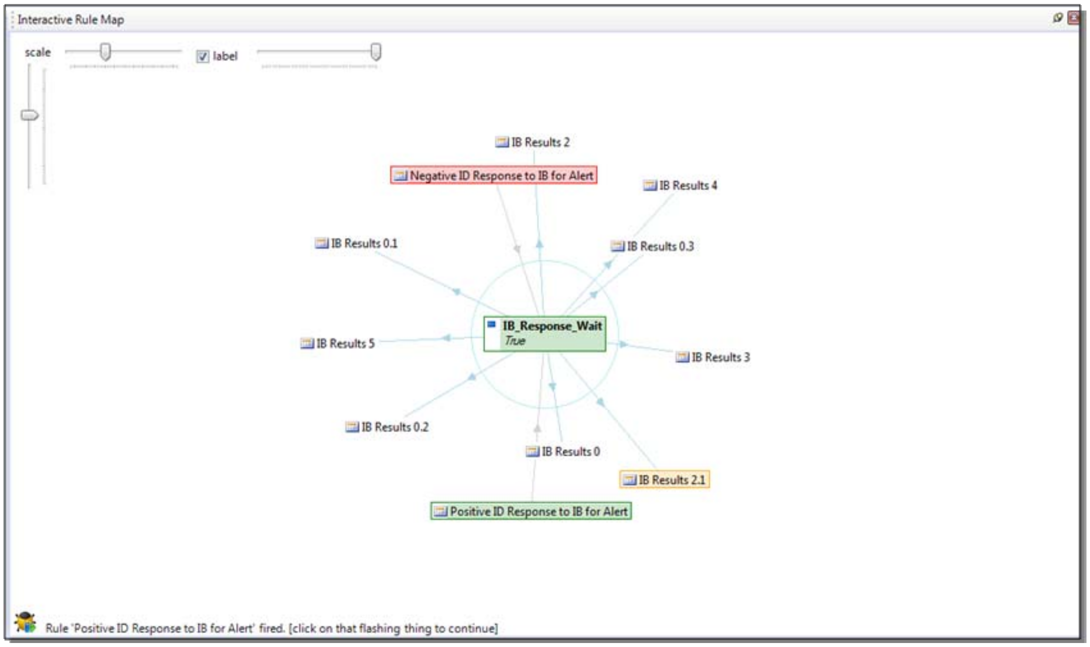
Rule 'Alert Notification is Present' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Alert Notification is Present
Alert Notification is Absent
Negative ID Response to IB for Alert
Query_Requires_Data_Insertion
True
Positive ID Response to IB for Alert

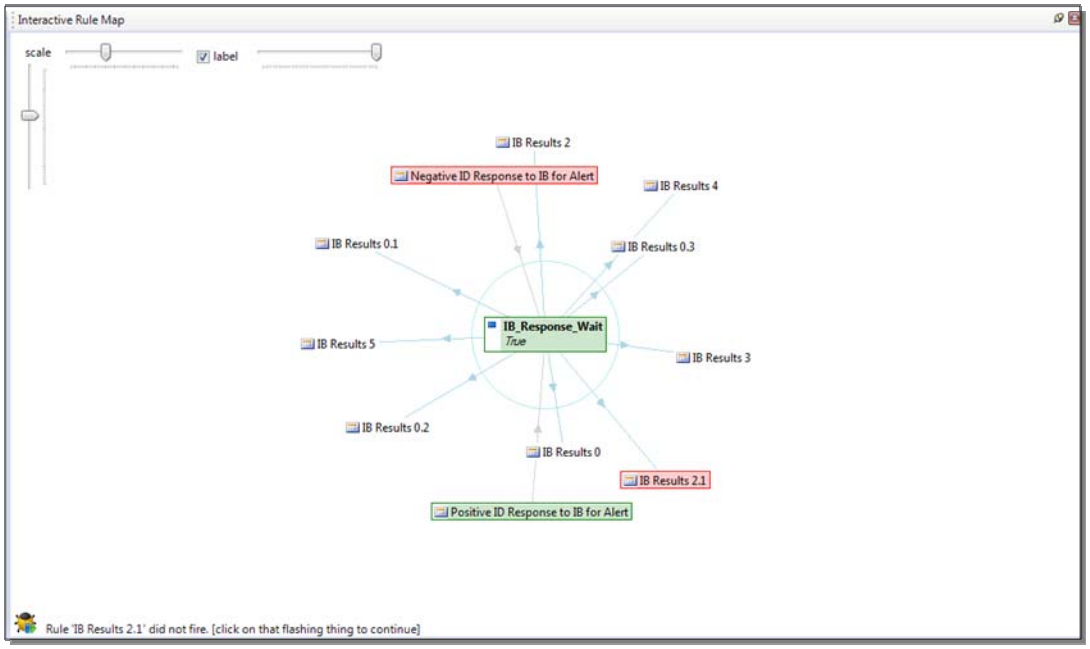
Rule 'Negative ID Response to IB for Alert' did not fire. [click on that flashing thing to continue]



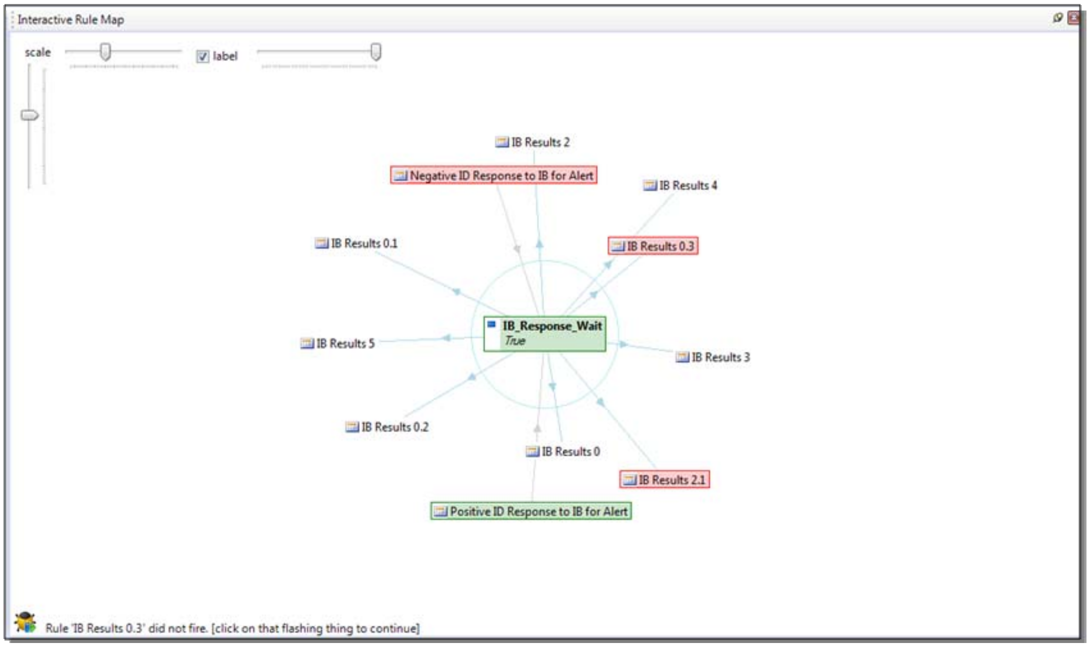
Rule 'Positive ID Response to IB for Alert' is in progress. [click on that flashing thing to continue]



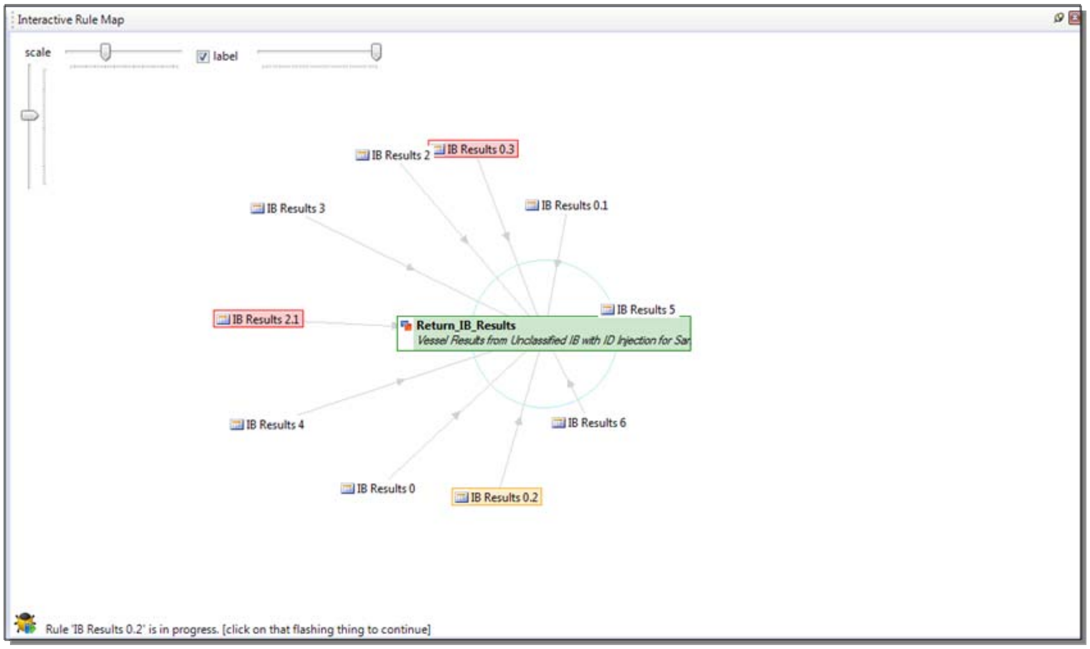
Rule 'Positive ID Response to IB for Alert' fired. [click on that flashing thing to continue]



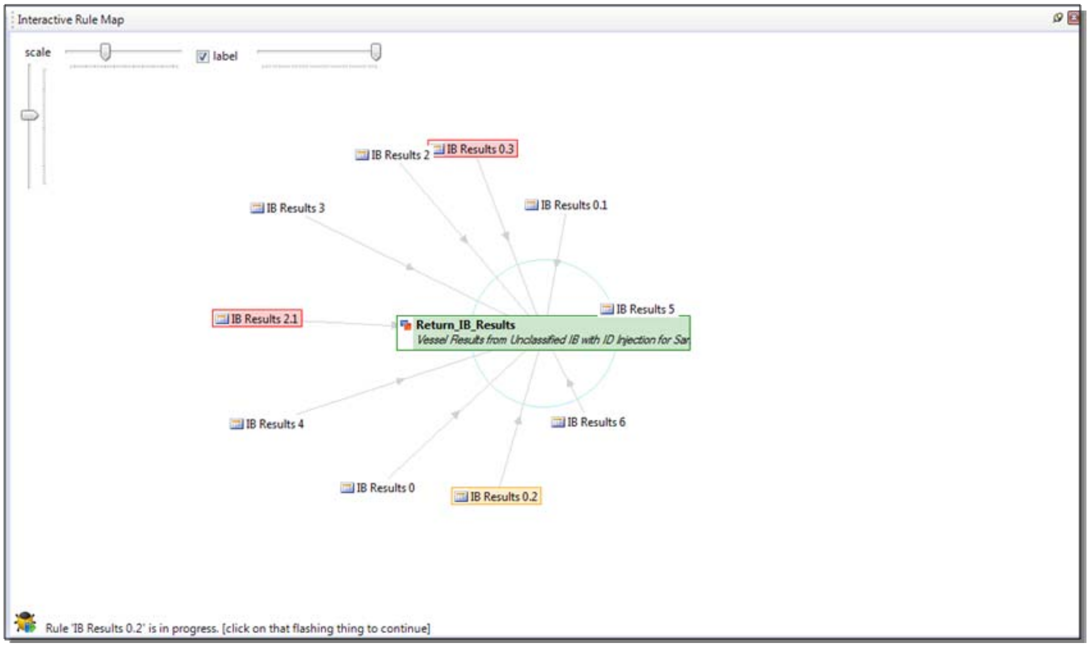
Rule 'IB Results 2.1' did not fire. [click on that flashing thing to continue]



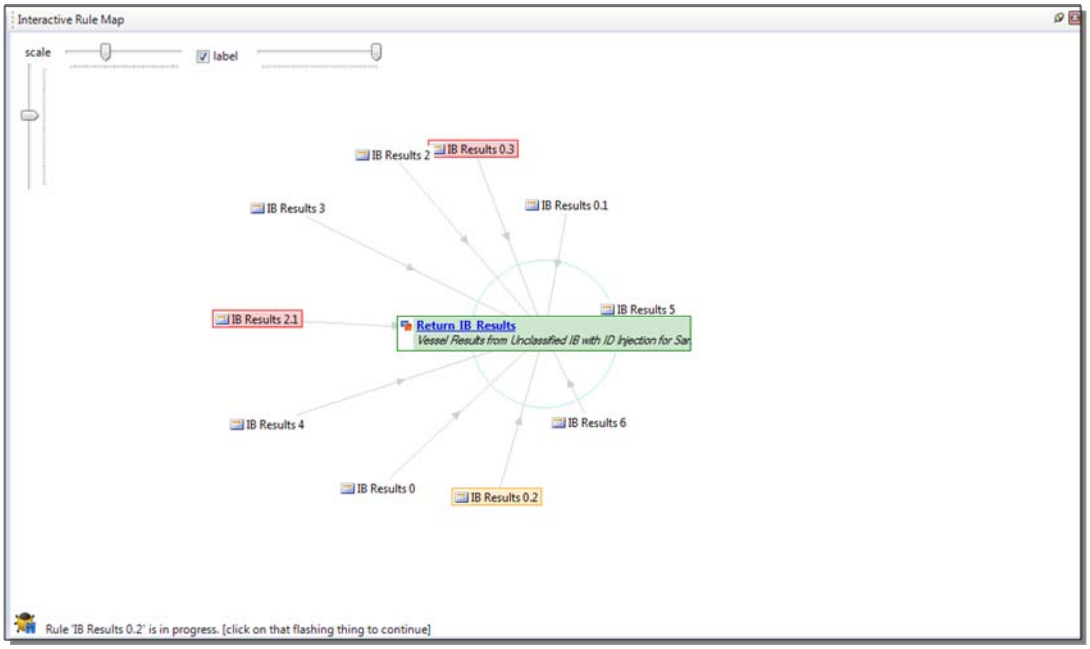
Rule 'IB Results 0.3' did not fire. [click on that flashing thing to continue]



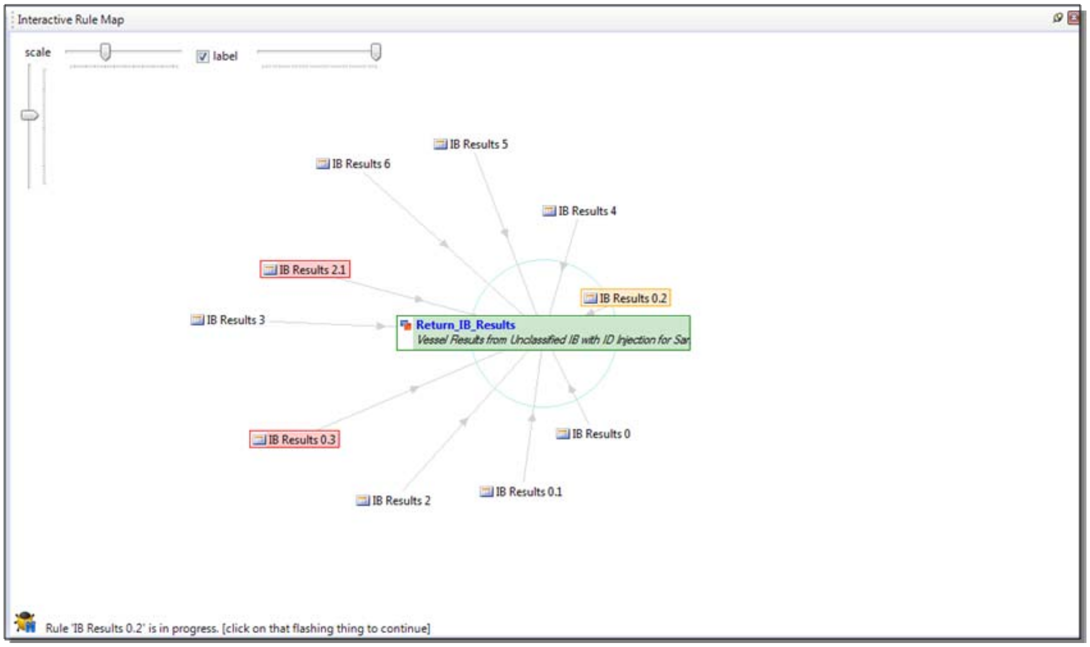
Rule 'IB Results 0.2' is in progress. [click on that flashing thing to continue]



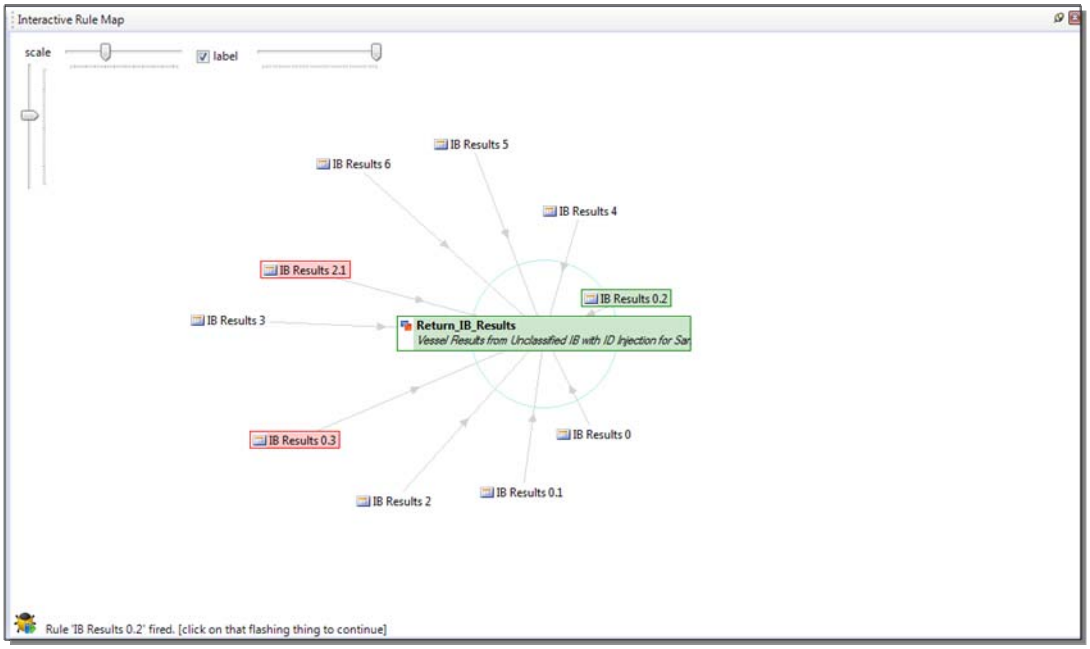
Rule 'IB Results 0.2' is in progress. [click on that flashing thing to continue]



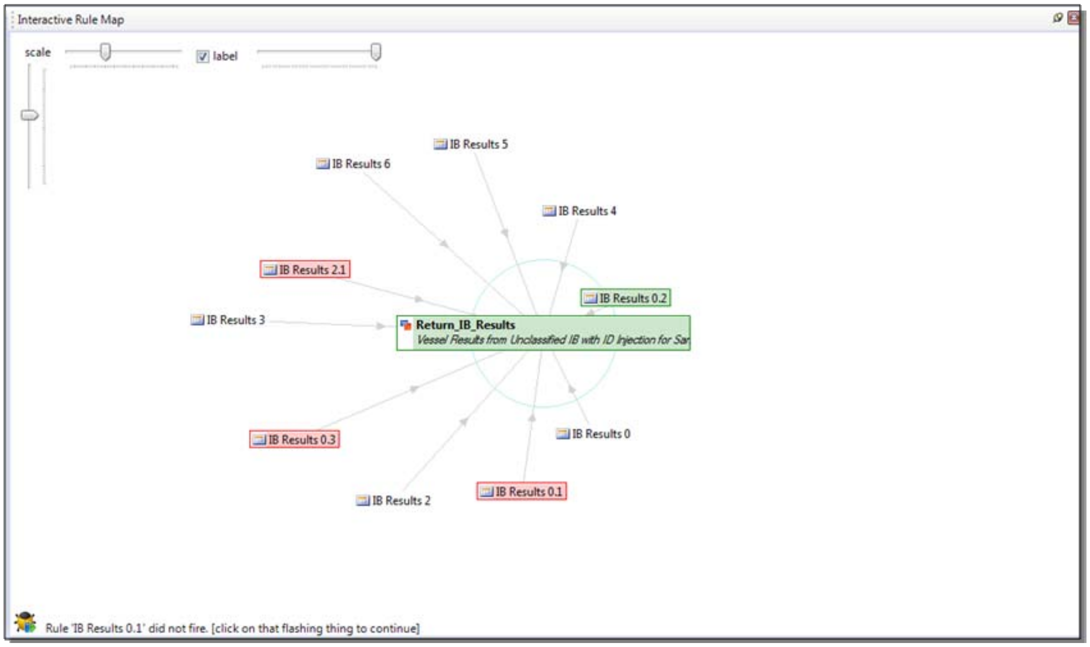
Rule 'IB Results 0.2' is in progress. [click on that flashing thing to continue]



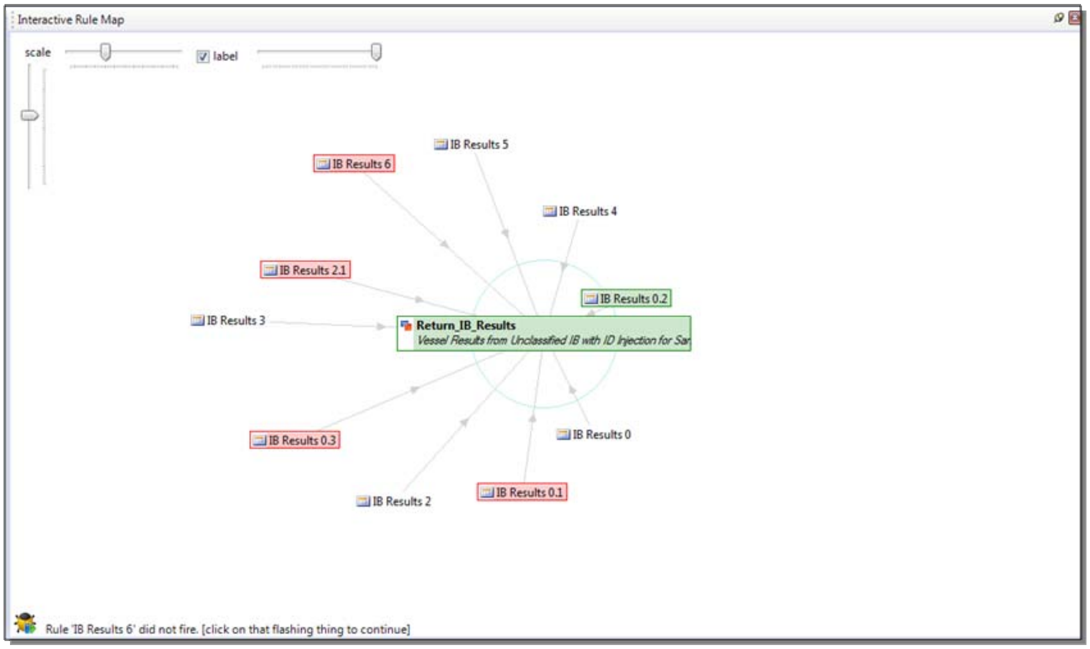
Rule 'IB Results 0.2' is in progress. [click on that flashing thing to continue]



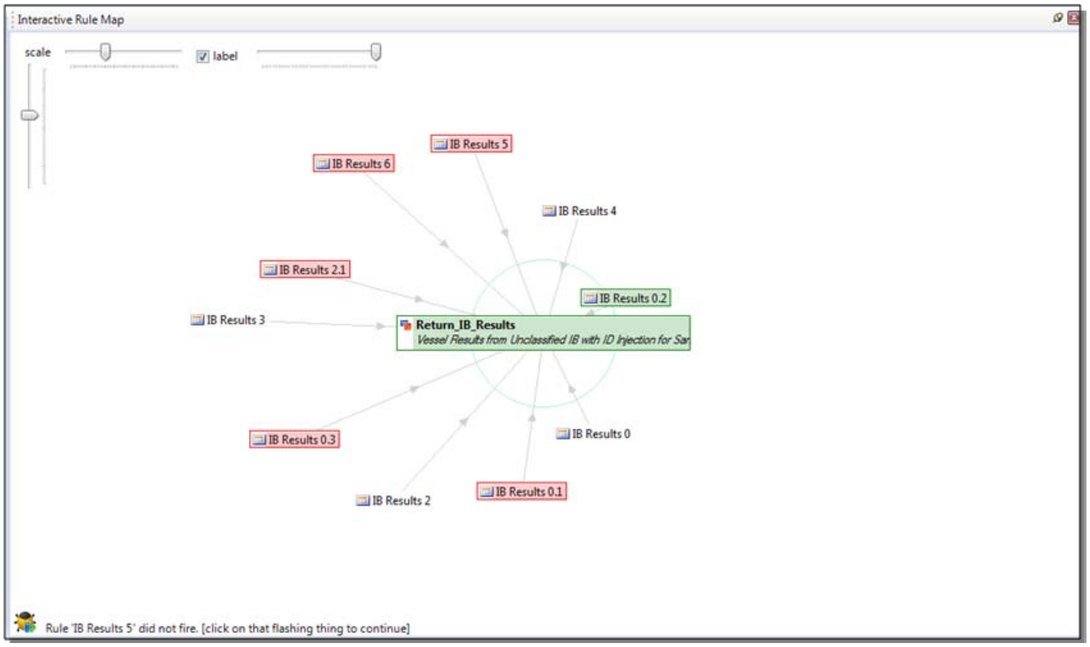
Rule 'IB Results 0.2' fired. [click on that flashing thing to continue]



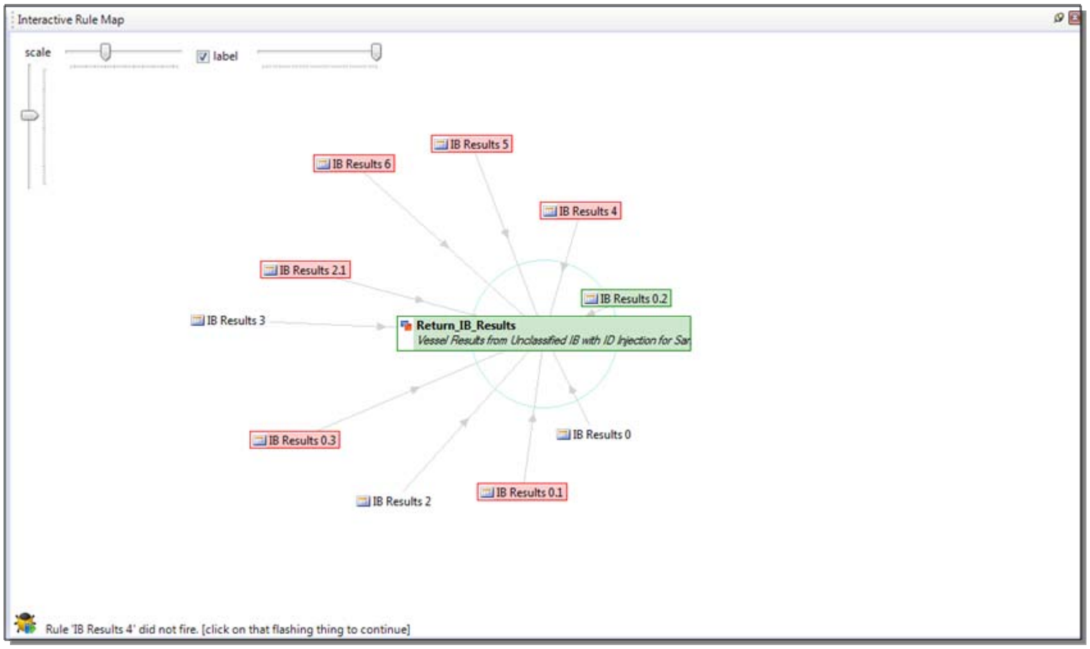
Rule 'IB Results 0.1' did not fire. [click on that flashing thing to continue]



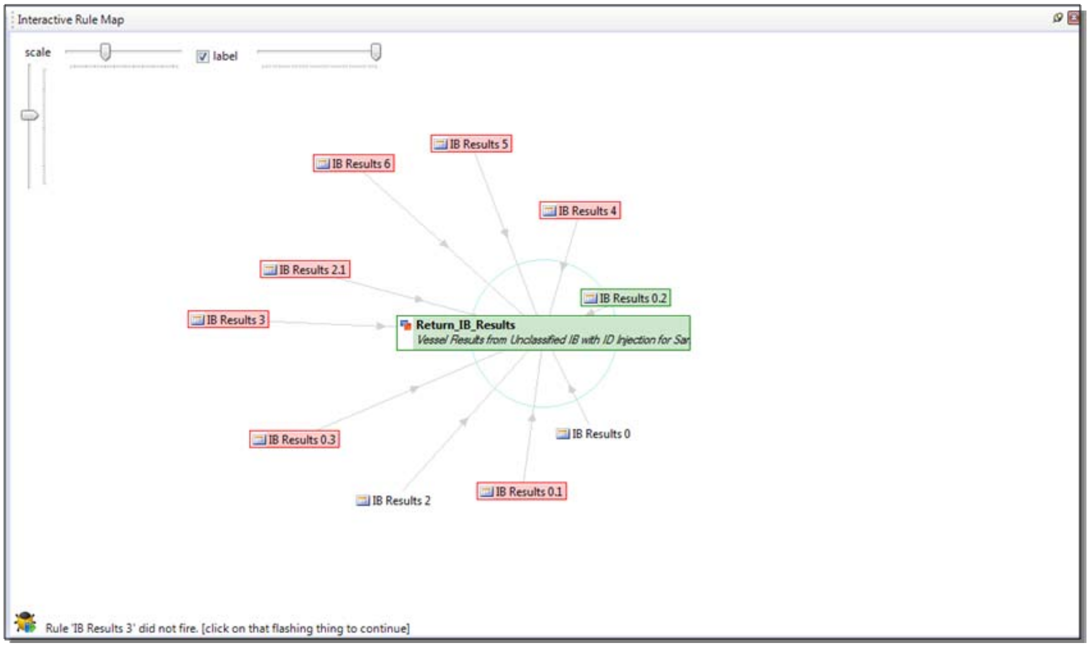
Rule 'IB Results 6' did not fire. [click on that flashing thing to continue]



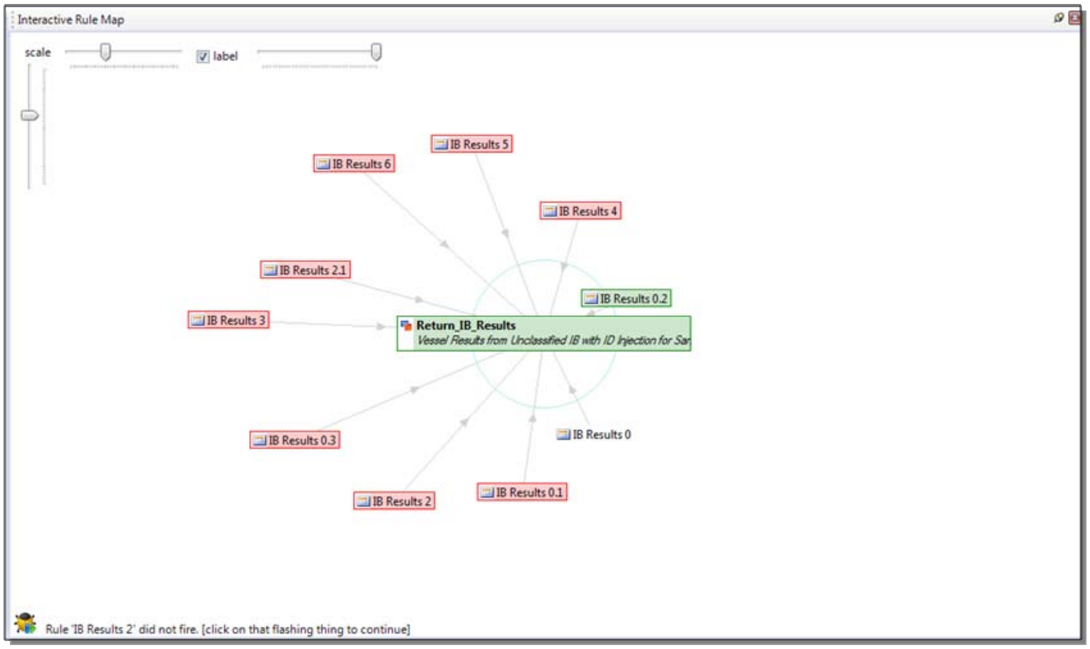
Rule 'IB Results 5' did not fire. [click on that flashing thing to continue]



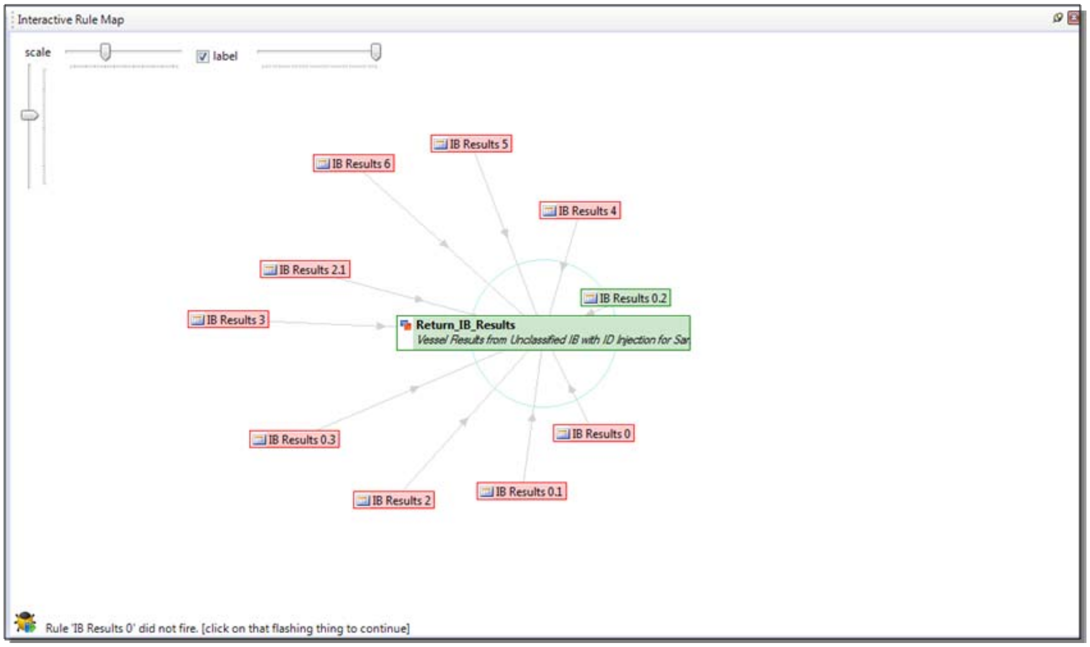
Rule 'IB Results 4' did not fire. [click on that flashing thing to continue]



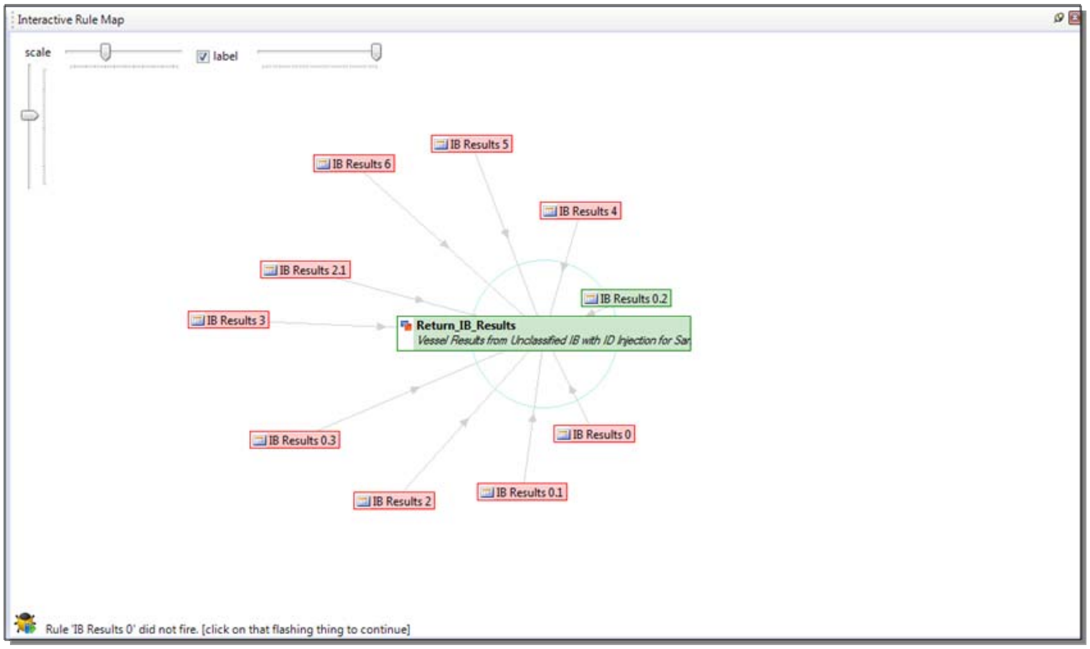
Rule 'IB Results 3' did not fire. [click on that flashing thing to continue]



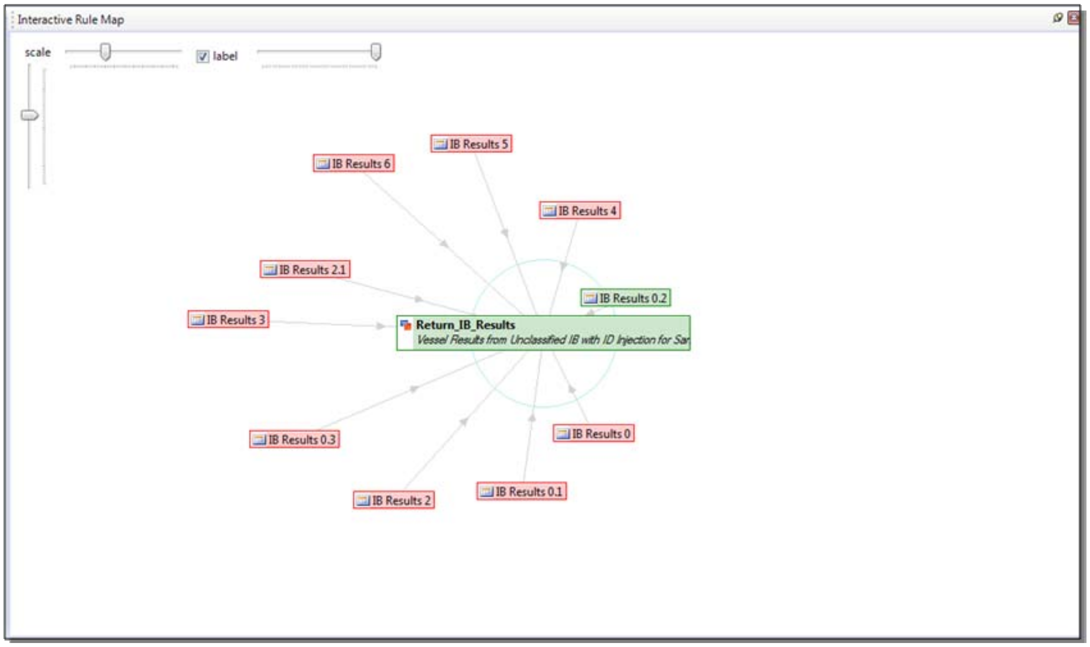
Rule 'IB Results 2' did not fire. [click on that flashing thing to continue]



Rule 'IB Results 0' did not fire. [click on that flashing thing to continue]



Rule 'IB Results 0' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
IB Results 5
IB Results 6
IB Results 4
IB Results 2.1
IB Results 0.2
IB Results 3
Return_IB_Results
Vessel Results from Unclassified IB with ID Injection for Sar
IB Results 0.3
IB Results 0
IB Results 2
IB Results 0.1

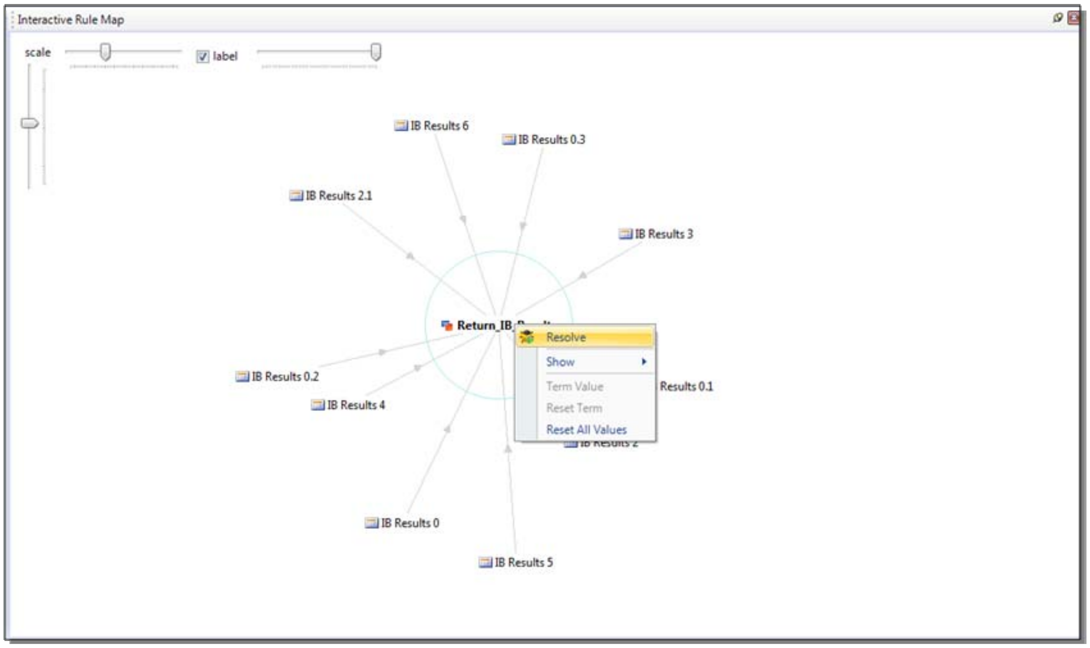
APPENDIX E. RULE TRACE OF OTHER QUERIES NOT SUPPORTED (INVALID QUERY)

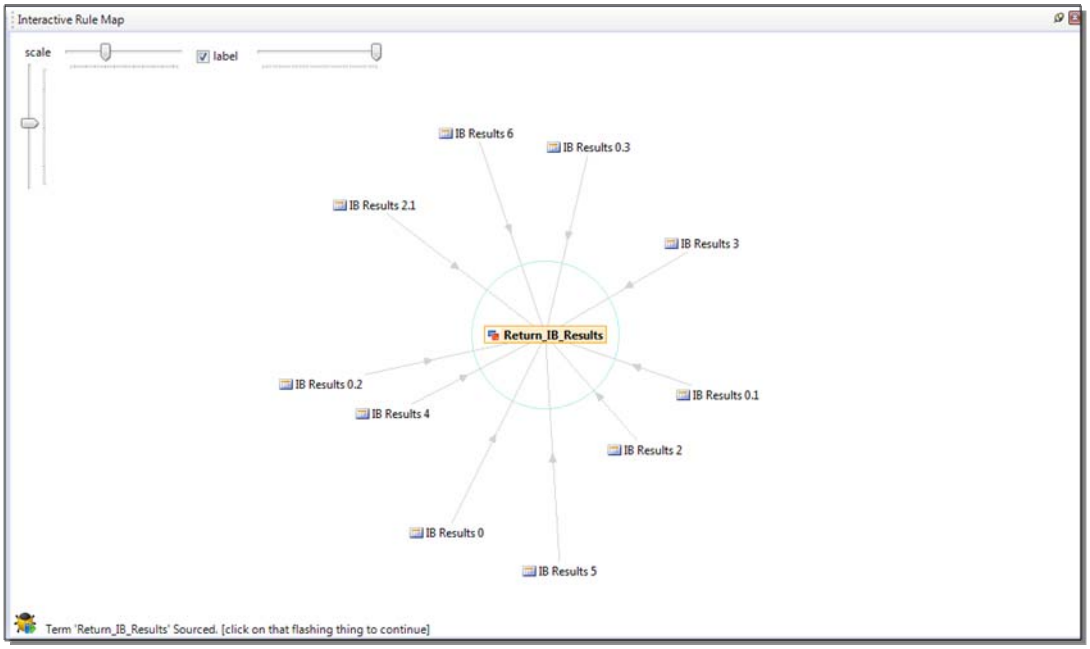
The visual trace of the ruleset execution was used to show non-support for actions that are not accounted for in the scenario and thus our ruleset. This is for items that are considered invalid by our security policy and should not return a valid result from the ruleset. This trace is completed using the following parameters:

Type of query:	Destination Port
Role / Actor:	Harbormaster
Location:	Shanghai
Security Level:	Unclassified
Alert Present:	False
Alert Classification Level:	Not Applicable
Expected Result:	“Invalid Query”

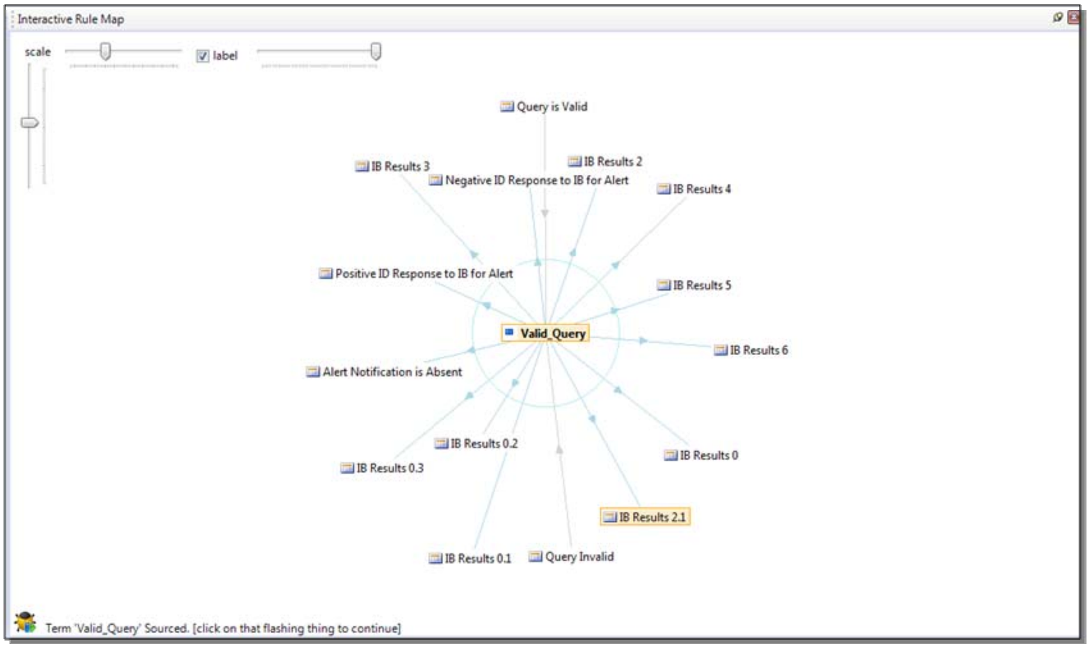
The expected result from this query and the RuleML execution is an IB response of “Invalid Query” to the user. The trace was completed using RuleManager’s Interactive Rule Map functionality. The pop-up boxes shown throughout the trace indicate the sourcing of predicates that would be included with tagged data in a live system. This was not replicated for this research and instead was manually inserted via the dialog boxes.

For the rule trace shown and from the interactive rule map of RuleManager, various indicators are used to show the actions during the trace. A rule or variable highlighted in Yellow, indicates that a rule or variable is currently being sourced. A rule depicted in Red indicates that the rule did not fire (execute) from the ruleset. A rule shown highlighted in Green indicates that the rule did fire and the result of that firing is also shown in the green highlighted box with the predicate name. Pop-up dialog boxes shown in the trace indicate the sourcing of a variable or predicate that would be done by an external entity (service) or taken from XML attributes attached to the query upon origination (i.e., user role and user security level).

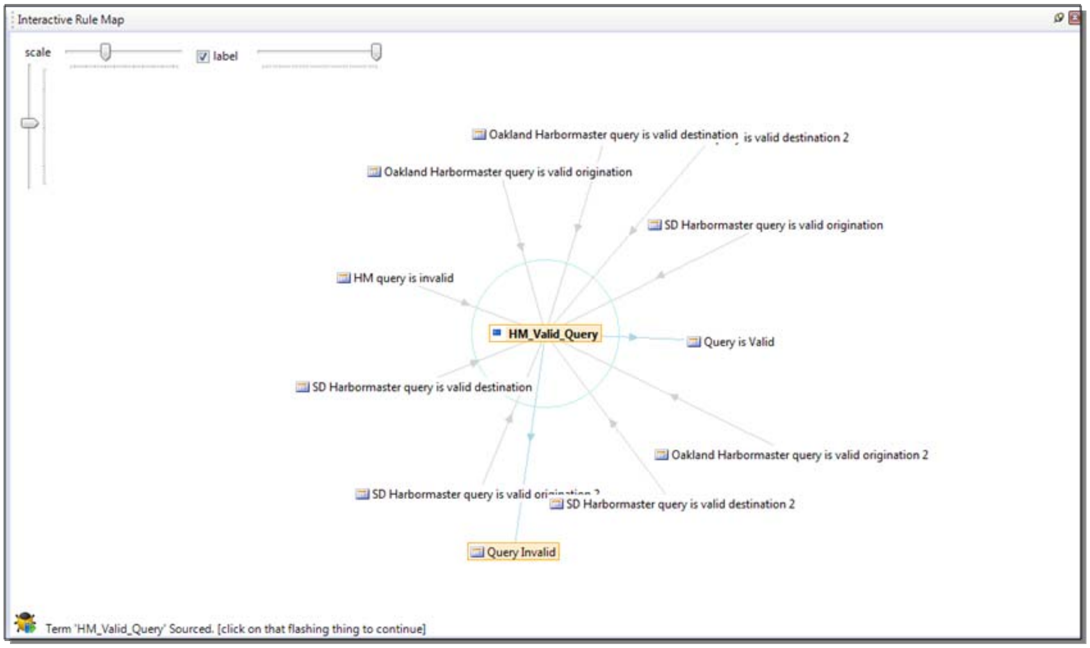




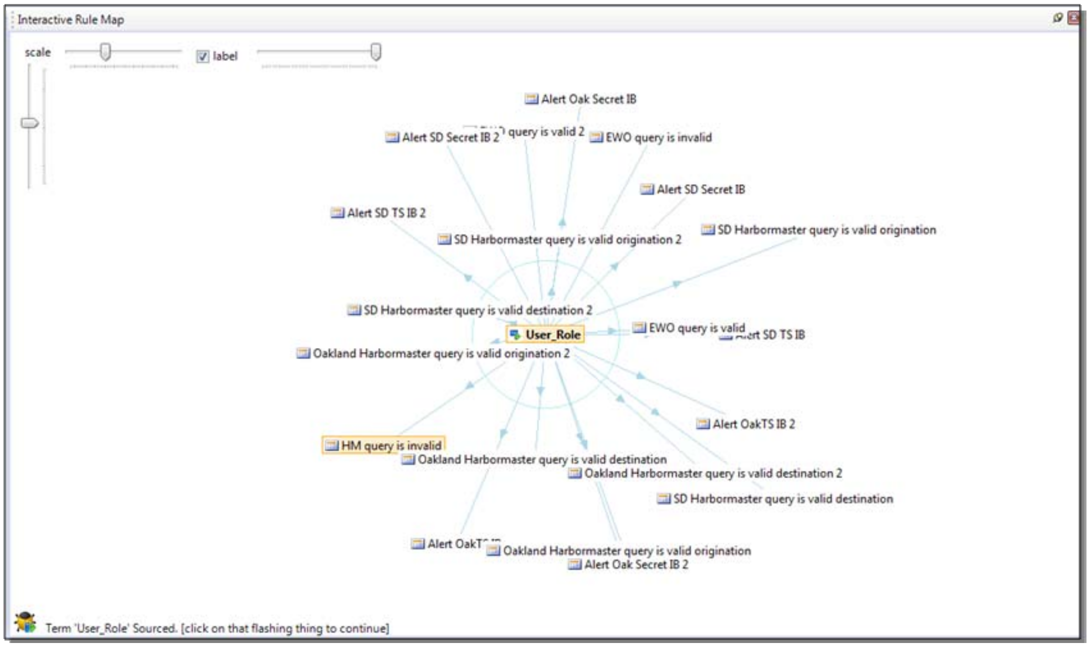
Term 'Return_IB_Results' Sourced. [click on that flashing thing to continue]



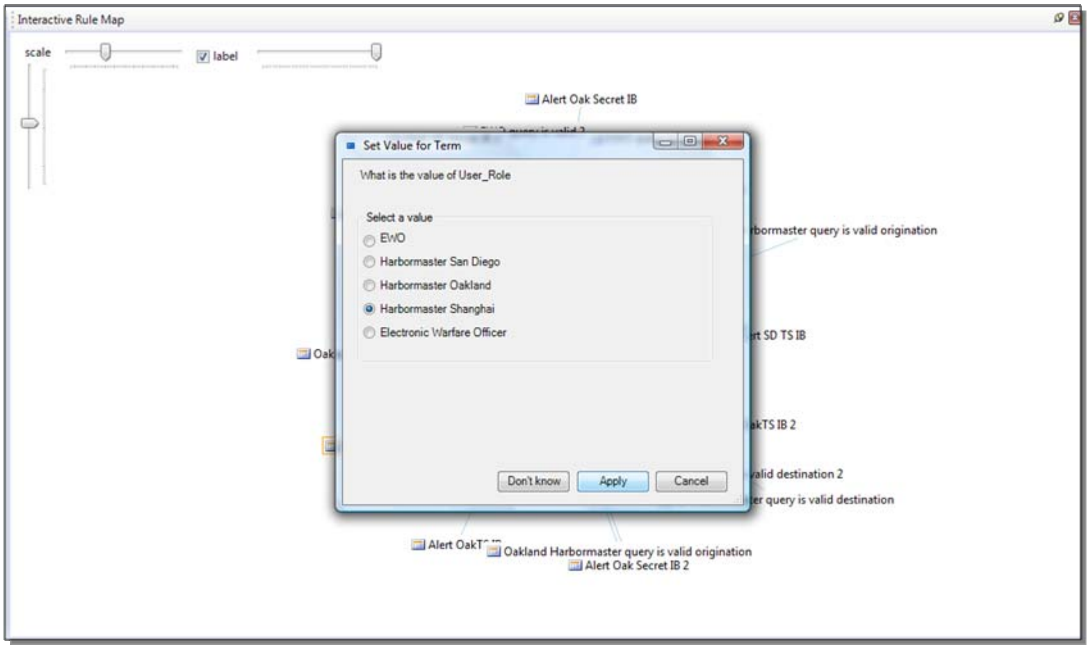
Term 'Valid_Query' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid destination
is valid destination 2
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid origination
HM query is invalid
HM_Valid_Query
Query is Valid
SD Harbormaster query is valid destination
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid destination 2
Query Invalid

Term 'HM_Valid_Query' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Alert Oak Secret IB
Alert SD Secret IB 2
query is valid 2
EWO query is invalid
Alert SD Secret IB
Alert SD TS IB 2
SD Harbormaster query is valid origination 2
SD Harbormaster query is valid origination
SD Harbormaster query is valid destination 2
User_Role
EWO query is valid
Alert SD TS IB
Oakland Harbormaster query is valid origination 2
Alert OakTS IB 2
HM query is invalid
Oakland Harbormaster query is valid destination
Oakland Harbormaster query is valid destination 2
SD Harbormaster query is valid destination
Alert OakT
Oakland Harbormaster query is valid origination
Alert Oak Secret IB 2

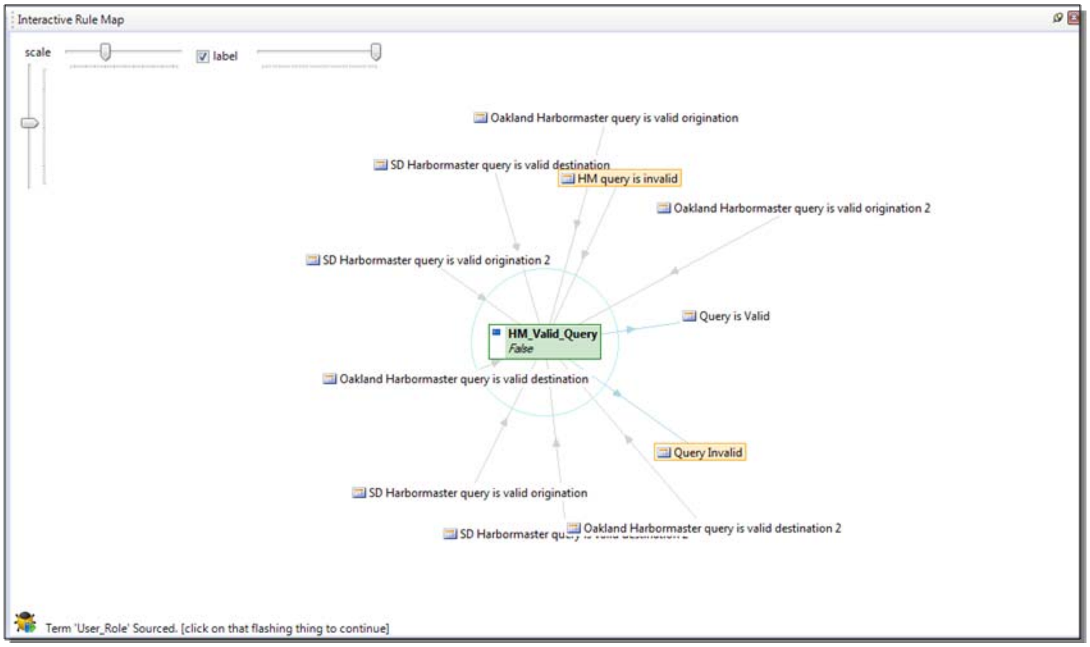
Term 'User_Role' Sourced. [click on that flashing thing to continue]



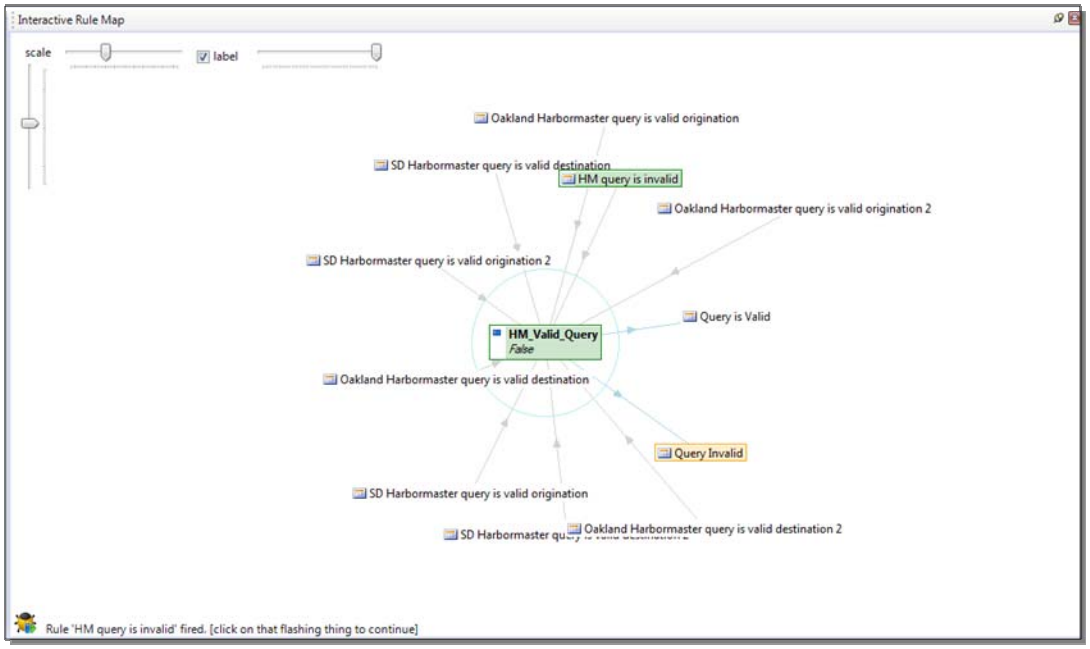
What is the value of User_Role

Select a value

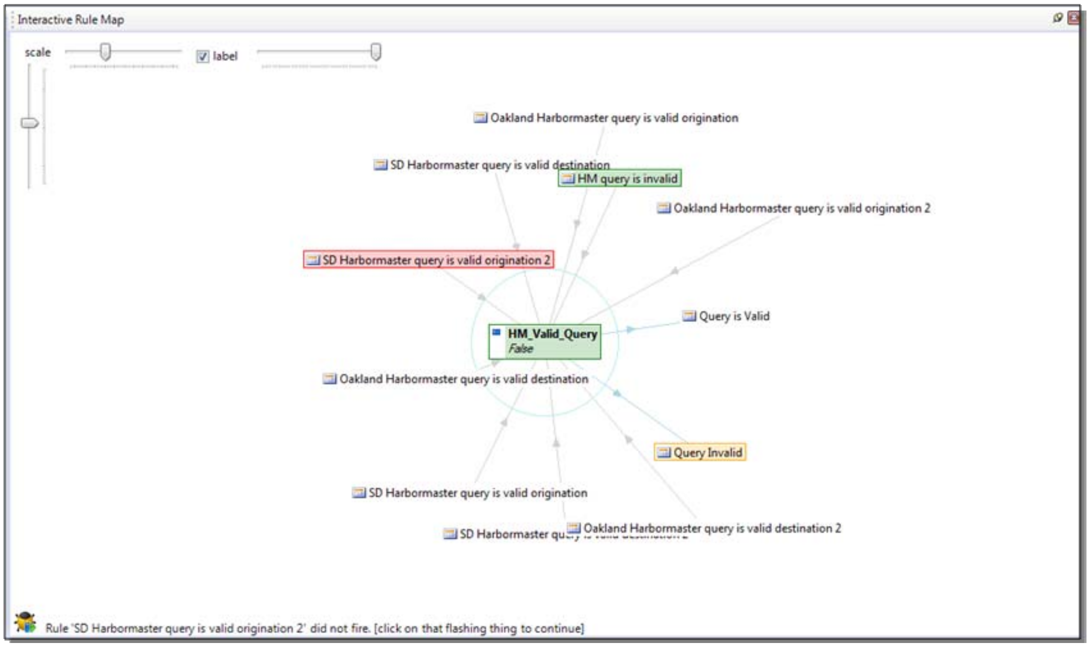
○ EWO
○ Harbormaster San Diego
○ Harbormaster Oakland
● Harbormaster Shanghai
○ Electronic Warfare Officer


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination 2

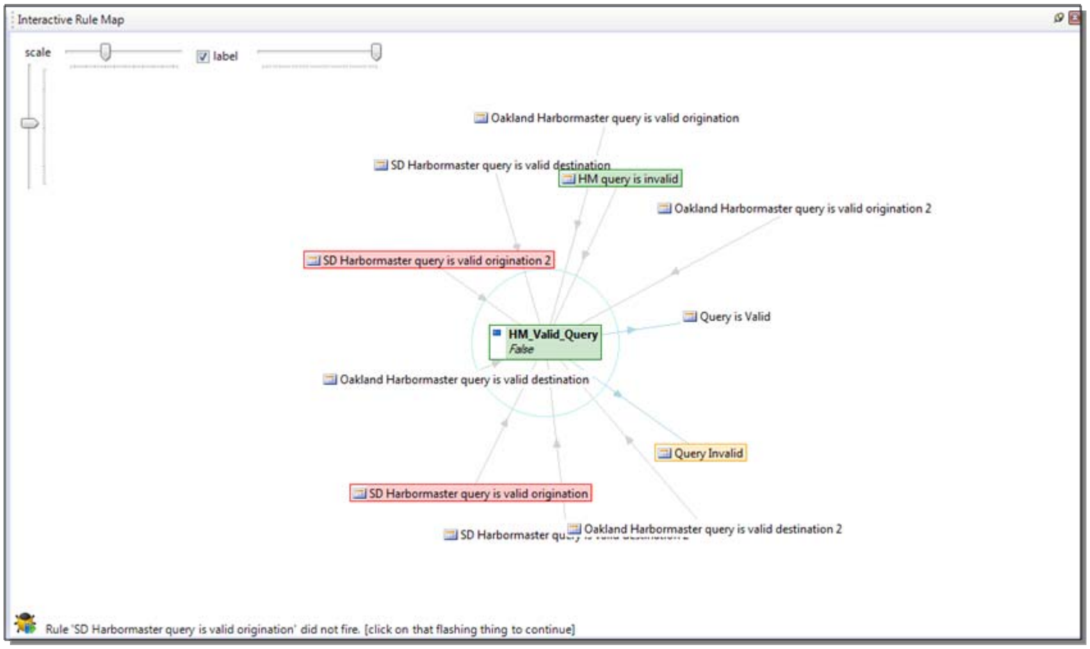
Term 'User_Role' Sourced. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination 2

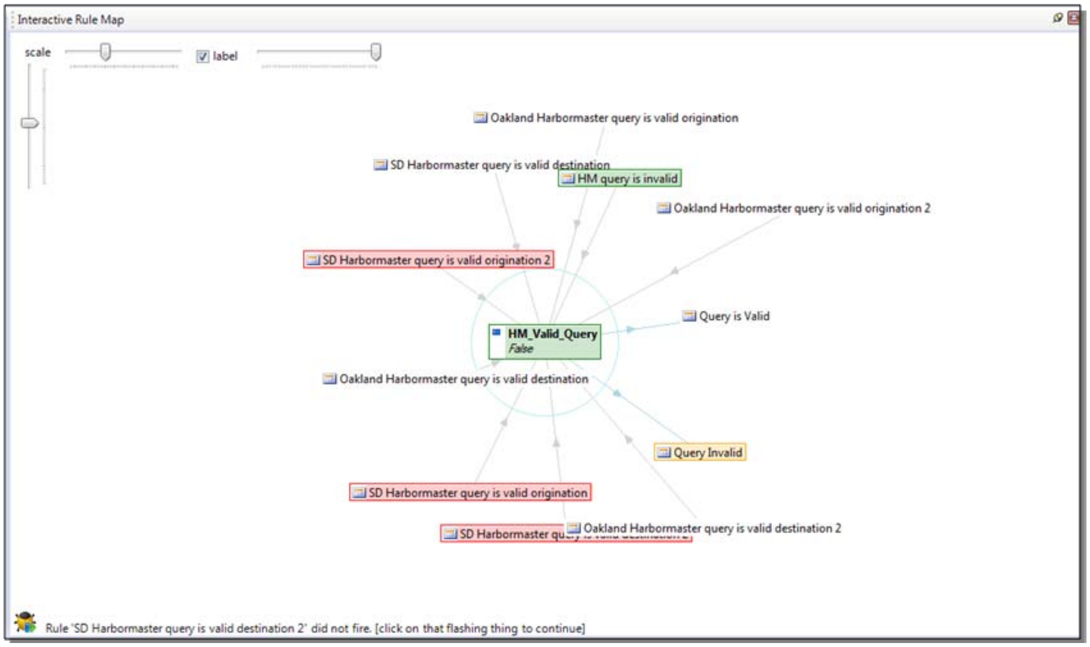
Rule 'HM query is invalid' fired. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster qu
Oakland Harbormaster query is valid destination 2

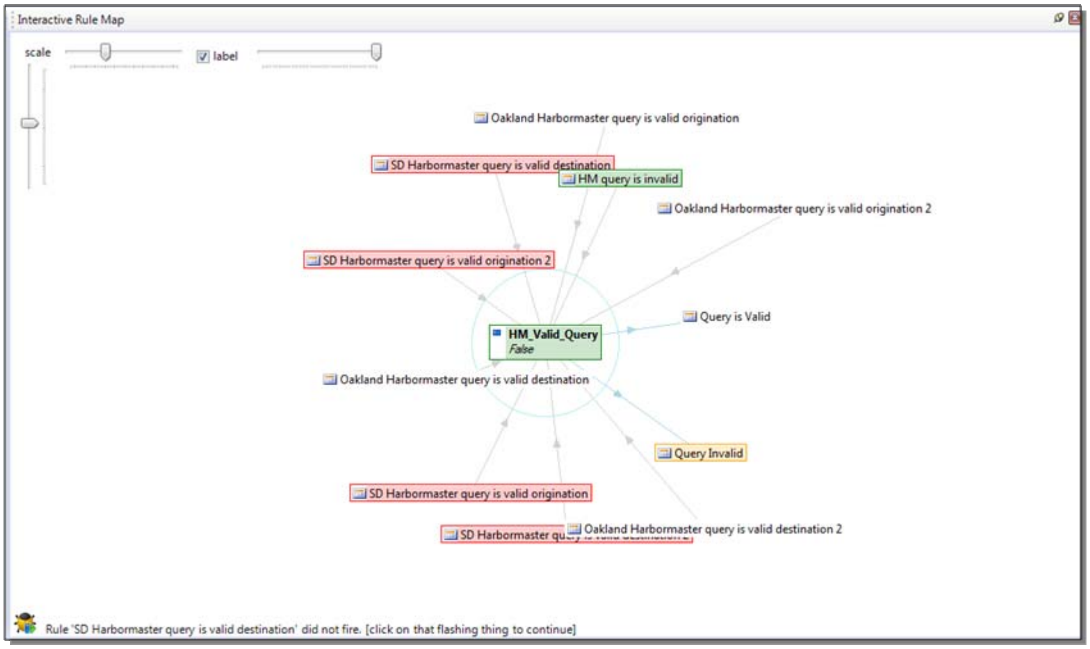
Rule 'SD Harbormaster query is valid origination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster qu
Oakland Harbormaster query is valid destination 2

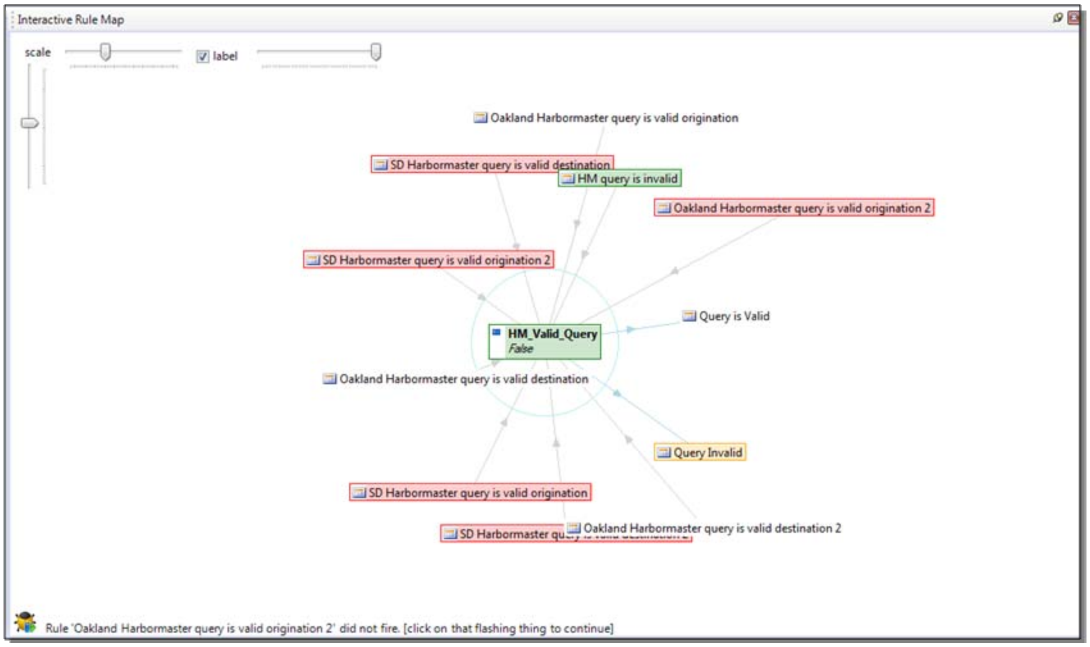
Rule 'SD Harbormaster query is valid origination' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster qu
Oakland Harbormaster query is valid destination 2

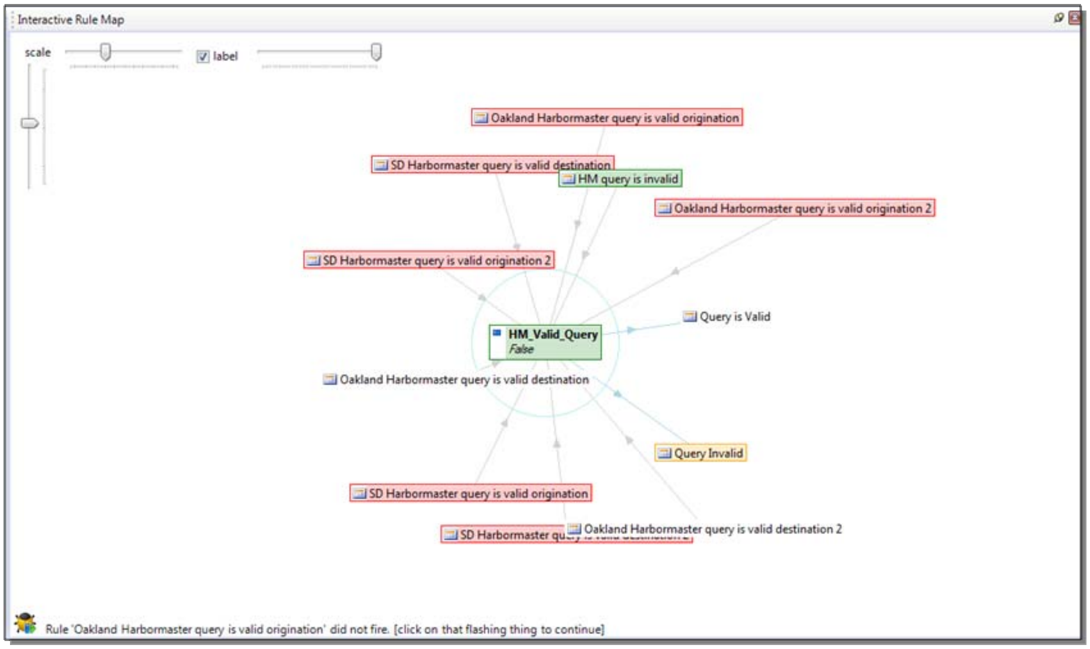
Rule 'SD Harbormaster query is valid destination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster qu
Oakland Harbormaster query is valid destination 2

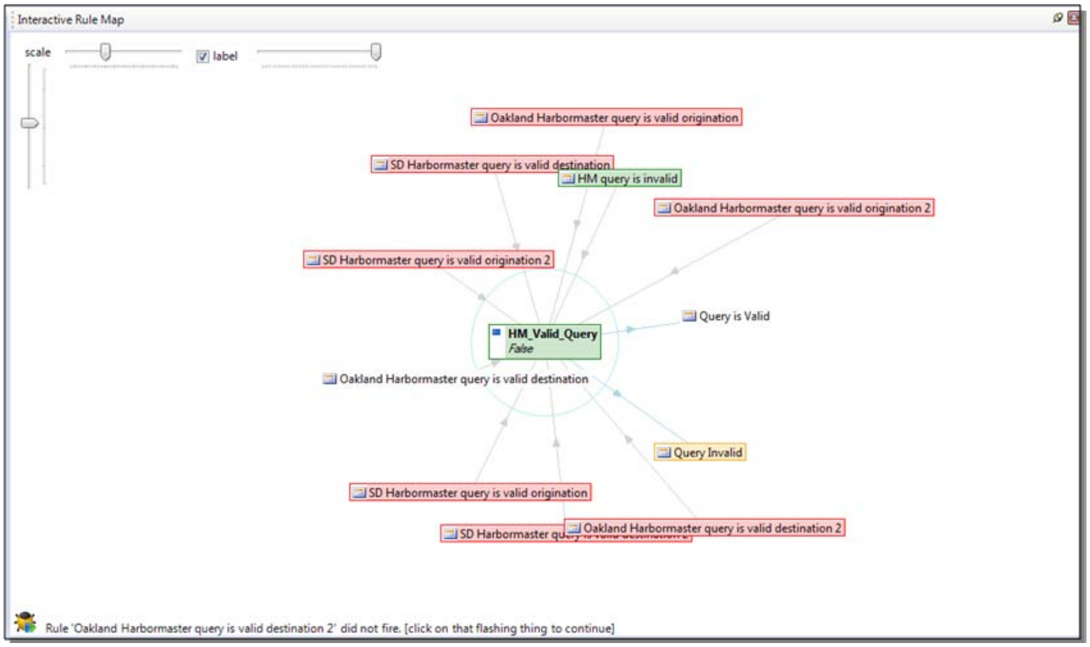
Rule 'SD Harbormaster query is valid destination' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination 2

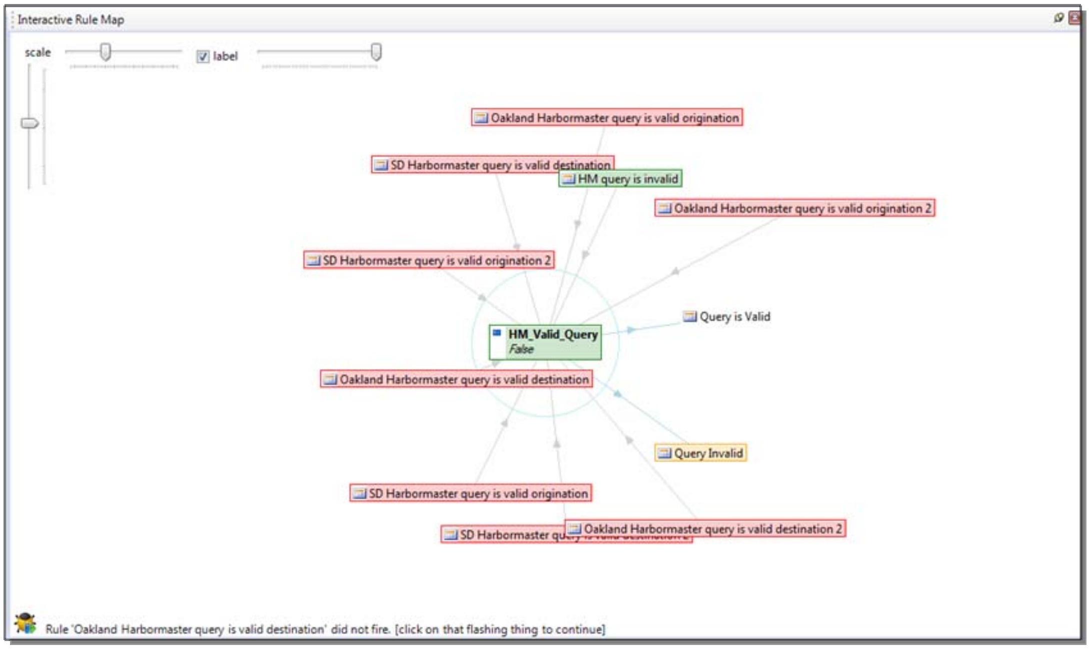
Rule 'Oakland Harbormaster query is valid origination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster query is valid destination 2
Oakland Harbormaster query is valid destination 2

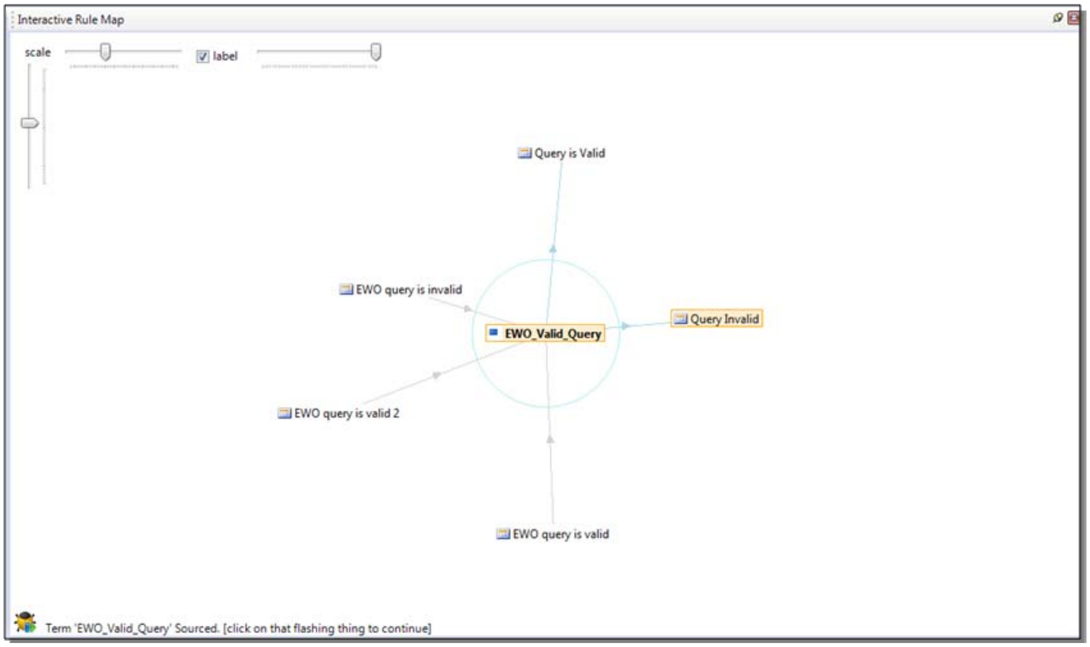
Rule 'Oakland Harbormaster query is valid origination' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster qu
Oakland Harbormaster query is valid destination 2

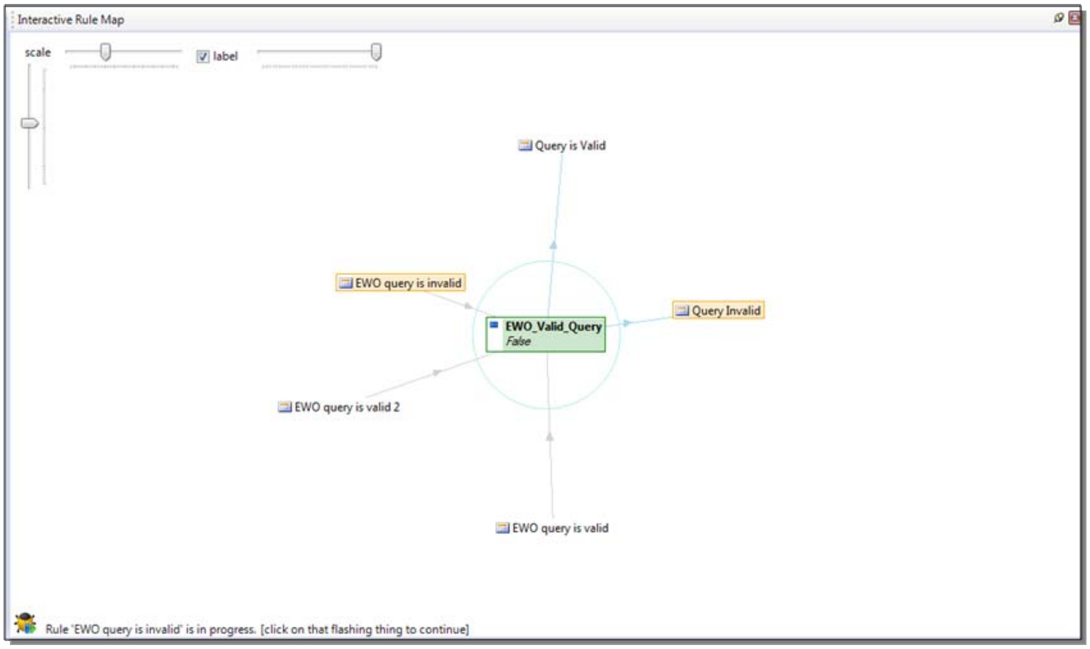
Rule 'Oakland Harbormaster query is valid destination 2' did not fire. [click on that flashing thing to continue]


Interactive Rule Map
scale
☑ label
Oakland Harbormaster query is valid origination
SD Harbormaster query is valid destination
HM query is invalid
Oakland Harbormaster query is valid origination 2
SD Harbormaster query is valid origination 2
Query is Valid
HM_Valid_Query
False
Oakland Harbormaster query is valid destination
Query Invalid
SD Harbormaster query is valid origination
SD Harbormaster qu
Oakland Harbormaster query is valid destination 2

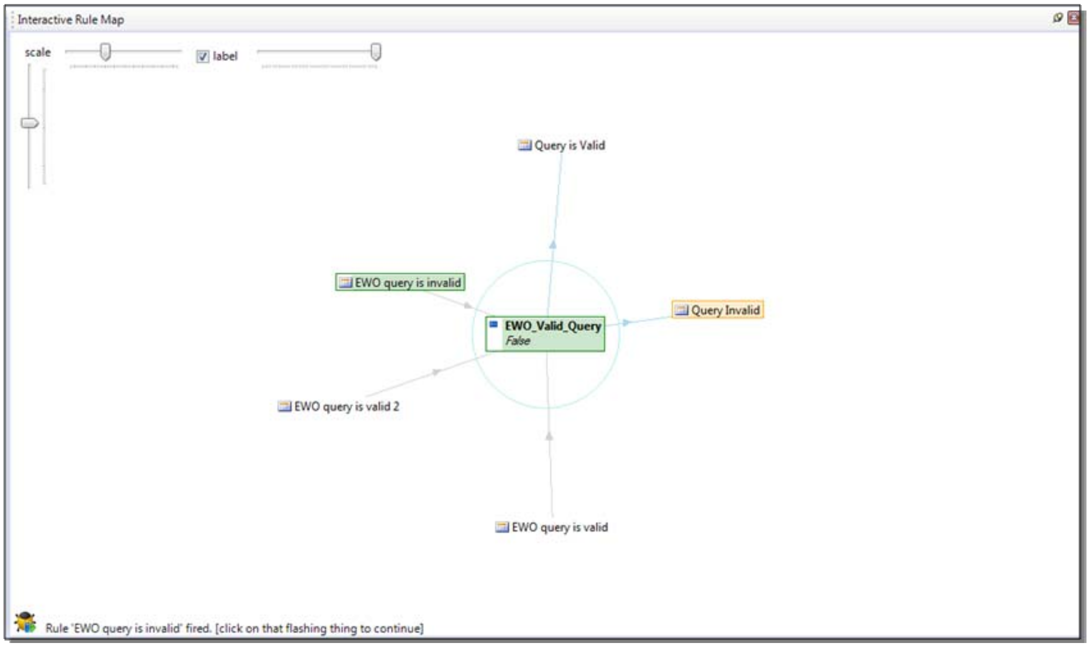
Rule 'Oakland Harbormaster query is valid destination' did not fire. [click on that flashing thing to continue]



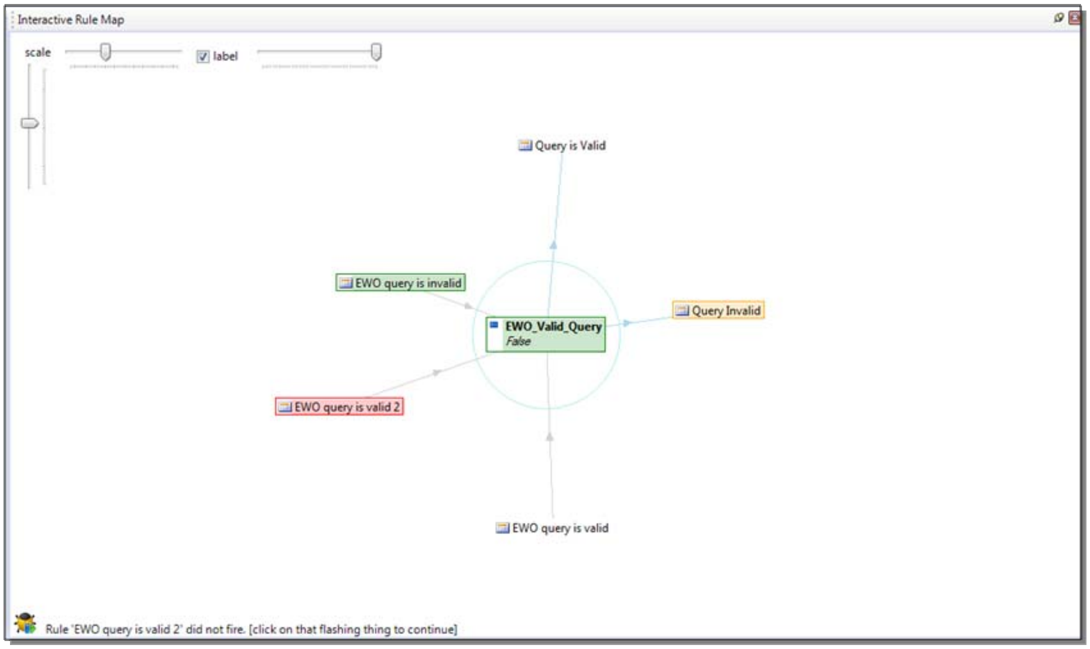
Term 'EWO_Valid_Query' Sourced. [click on that flashing thing to continue]



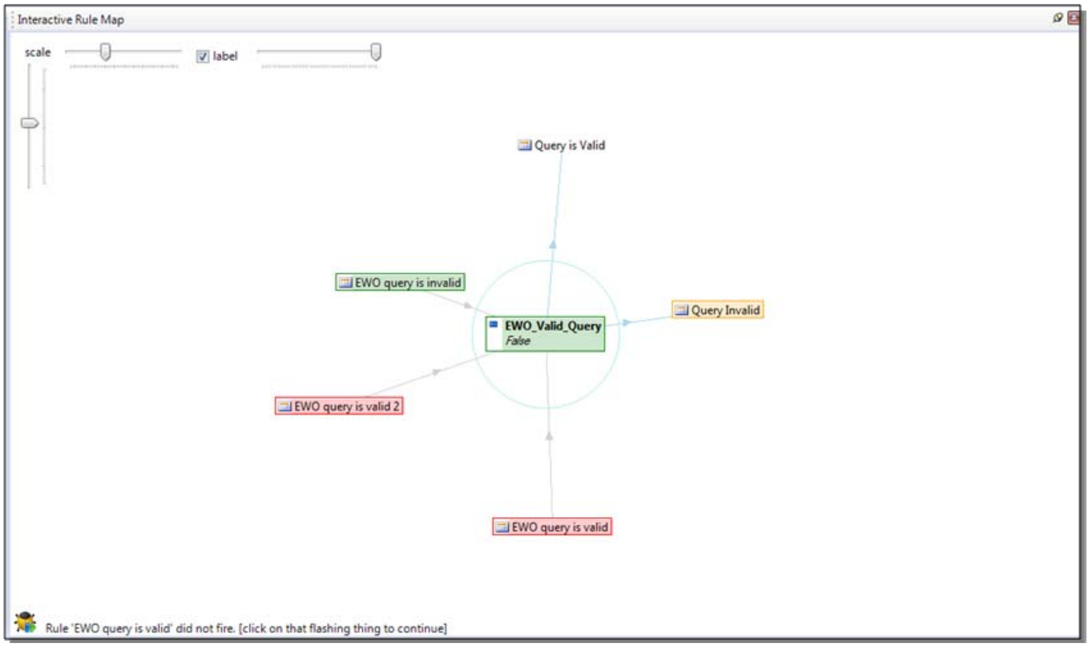
Rule 'EWO query is invalid' is in progress. [click on that flashing thing to continue]



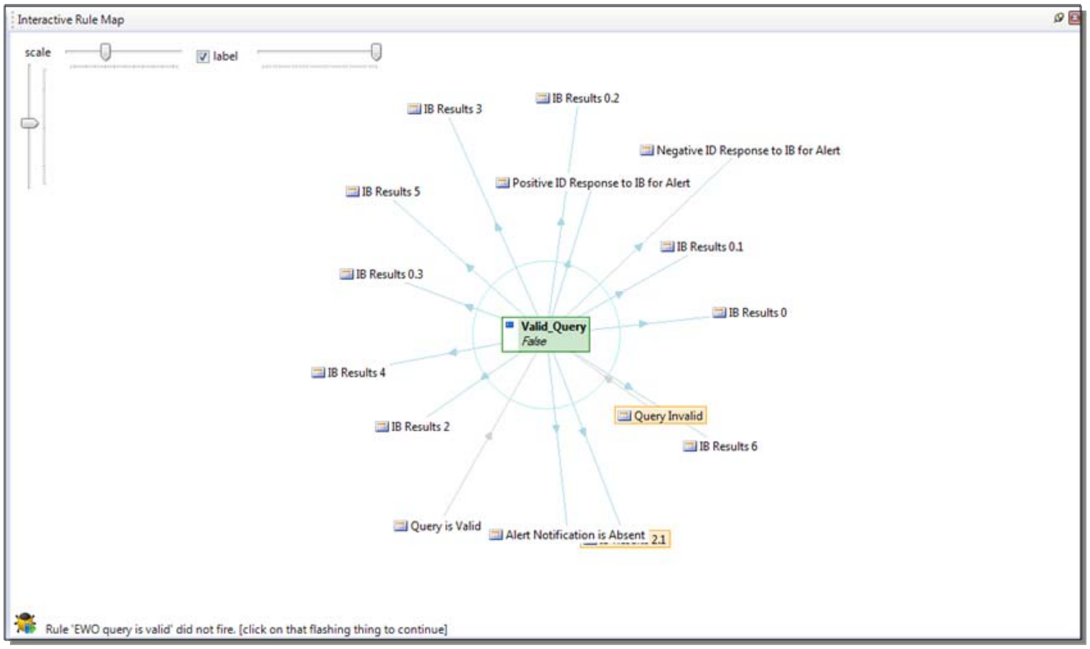
Rule 'EWO query is invalid' fired. [click on that flashing thing to continue]



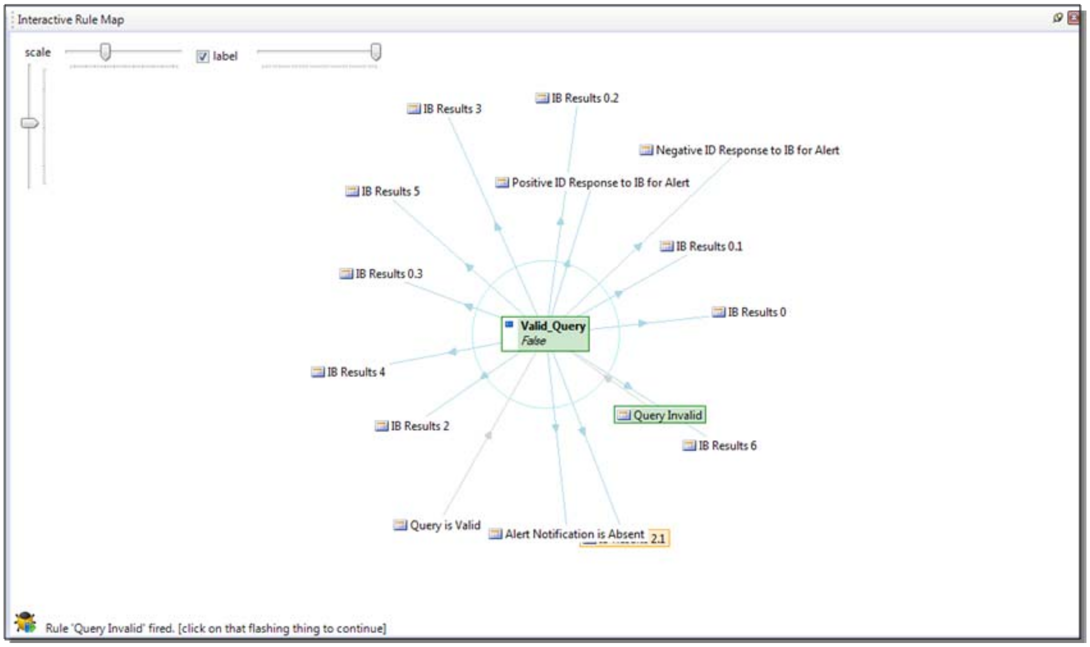
Rule 'EWO query is valid 2' did not fire. [click on that flashing thing to continue]



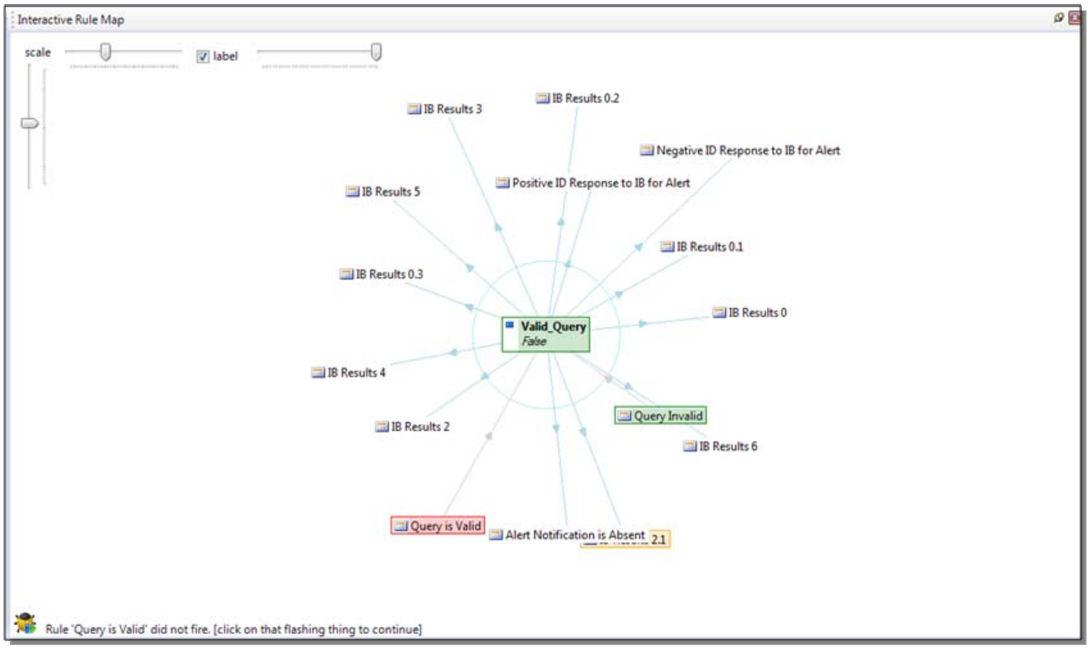
Rule 'EWO query is valid' did not fire. [click on that flashing thing to continue]



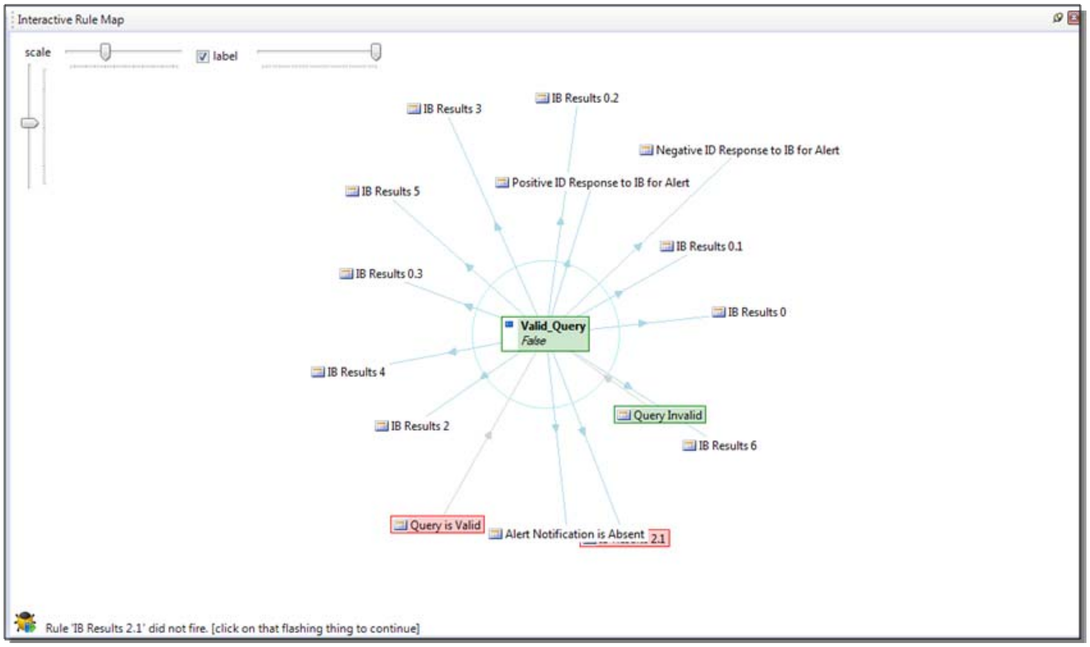
Rule 'EWO query is valid' did not fire. [click on that flashing thing to continue]



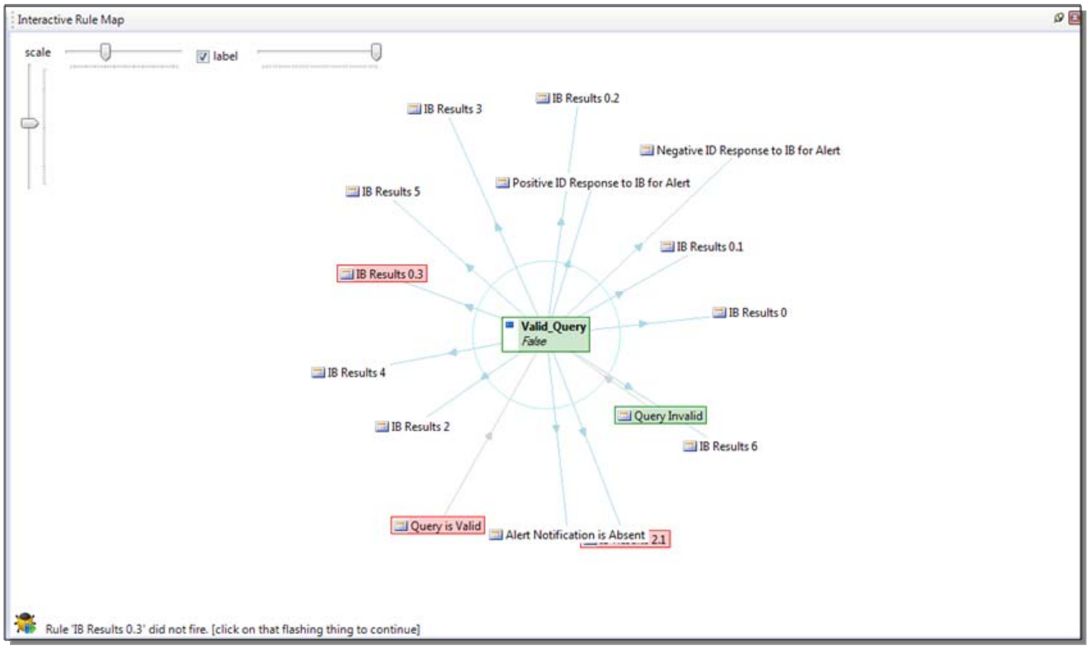
Rule 'Query Invalid' fired. [click on that flashing thing to continue]



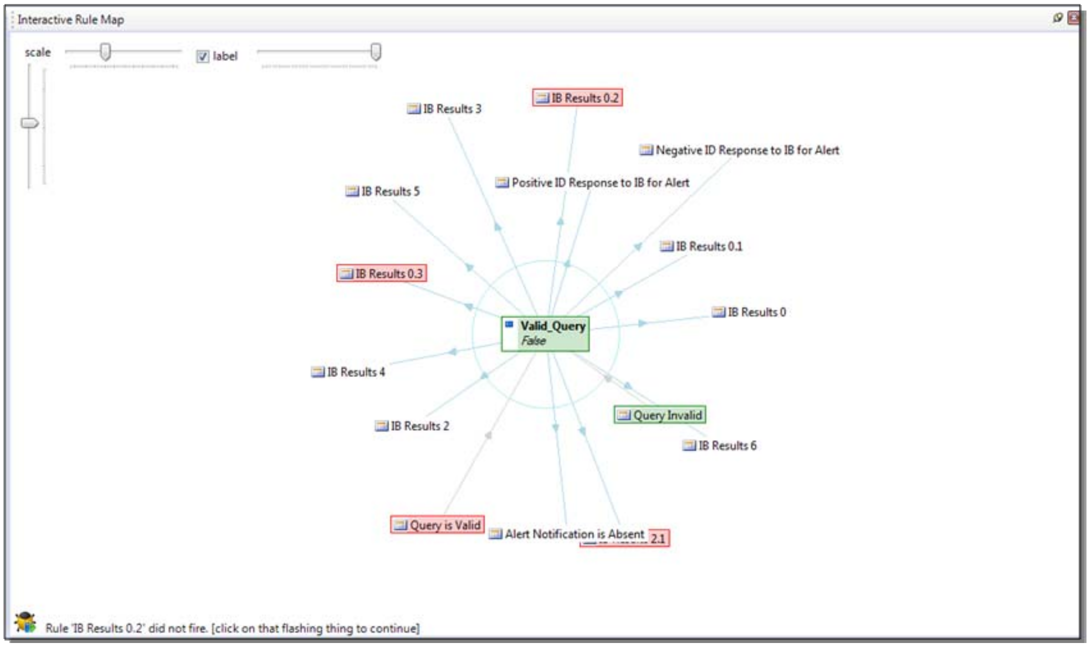
Rule 'Query is Valid' did not fire. [click on that flashing thing to continue]



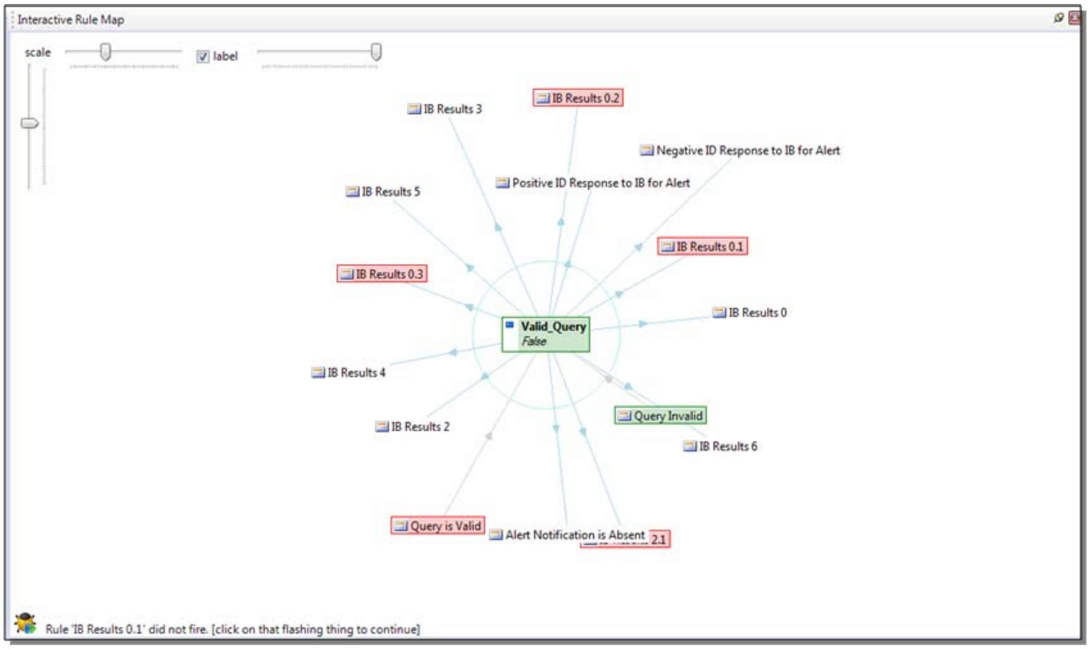
Rule 'IB Results 2.1' did not fire. [click on that flashing thing to continue]



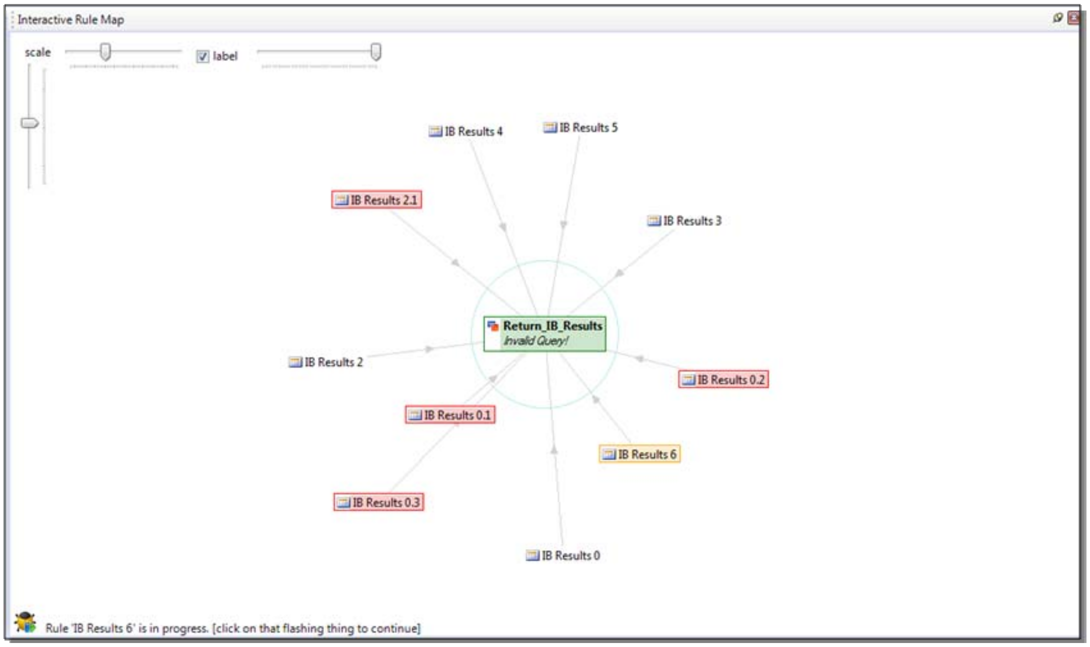
Rule 'IB Results 0.3' did not fire. [click on that flashing thing to continue]



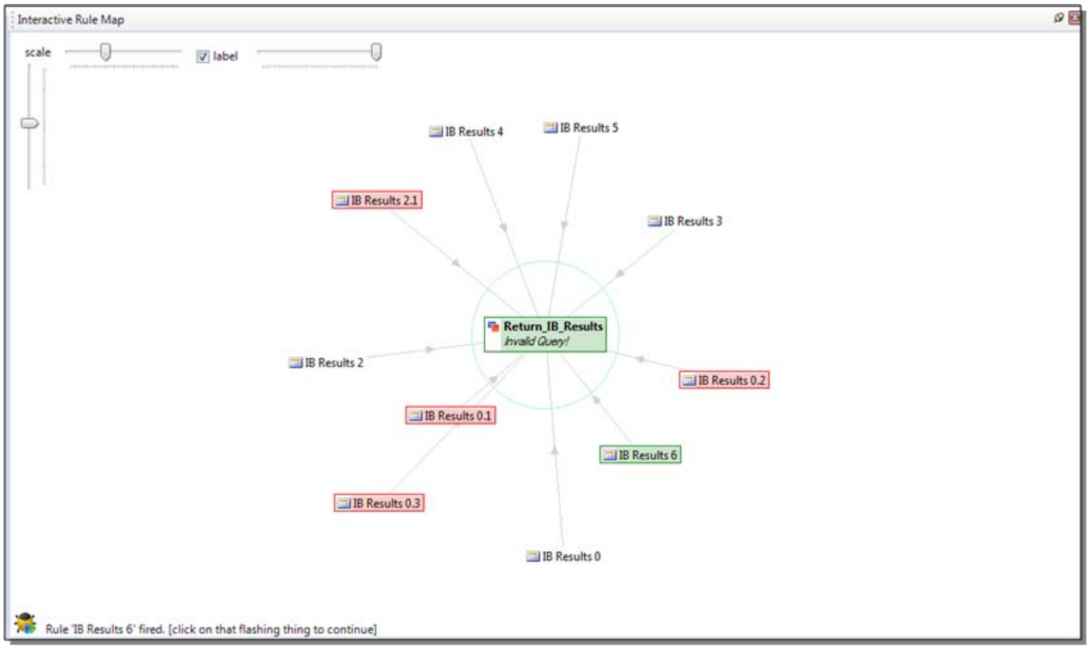
Rule 'IB Results 0.2' did not fire. [click on that flashing thing to continue]



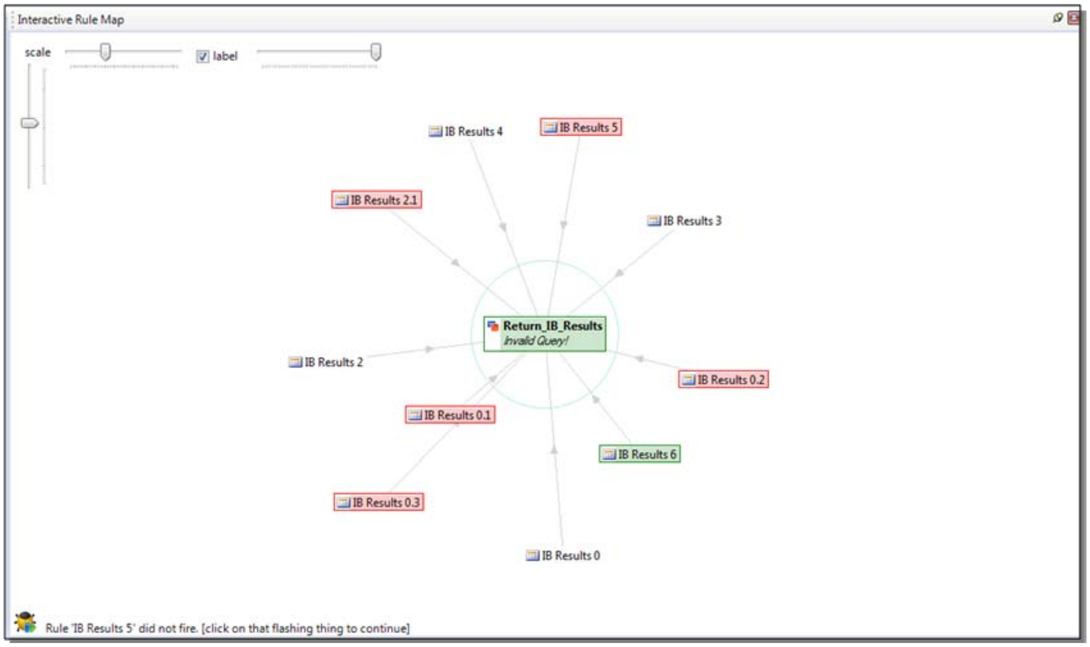
Rule 'IB Results 0.1' did not fire. [click on that flashing thing to continue]



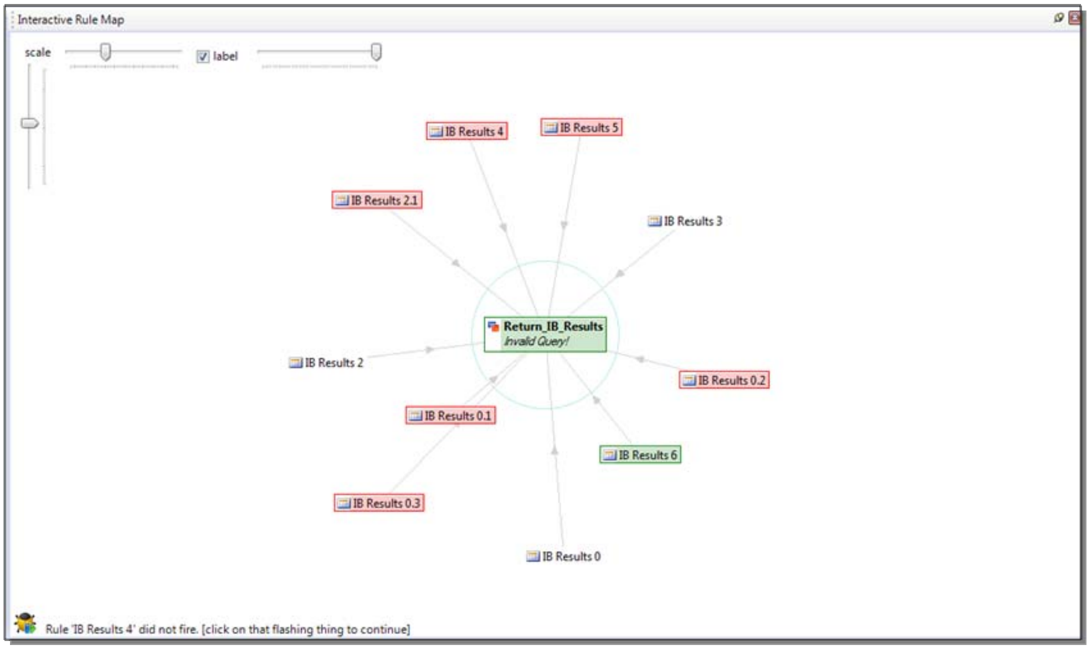
Rule 'IB Results 6' is in progress. [click on that flashing thing to continue]



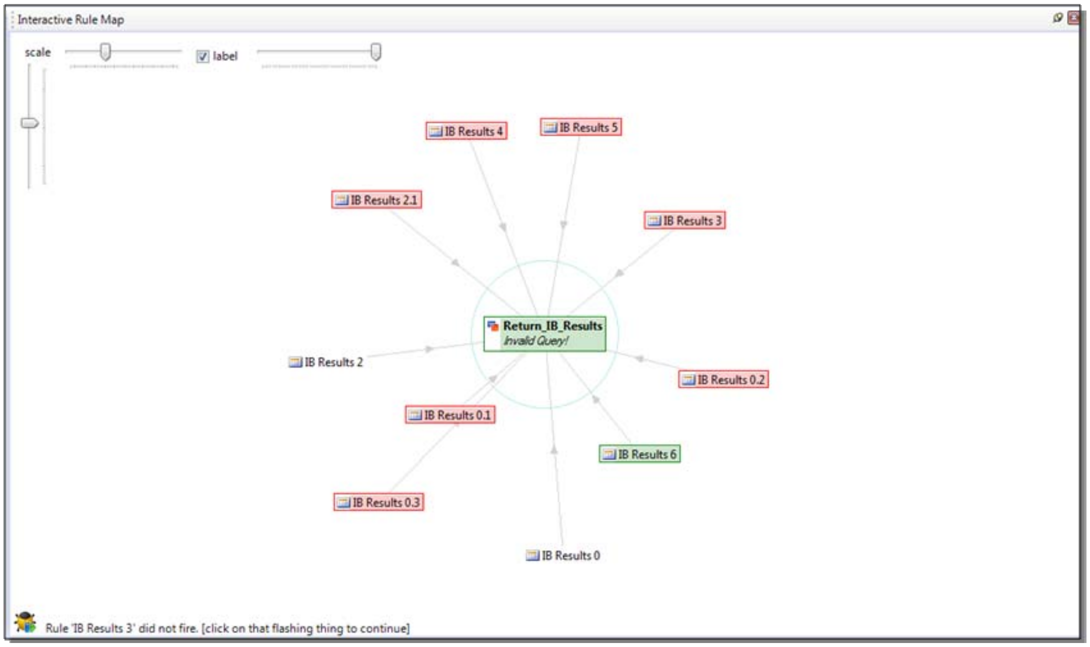
Rule 'IB Results 6' fired. [click on that flashing thing to continue]



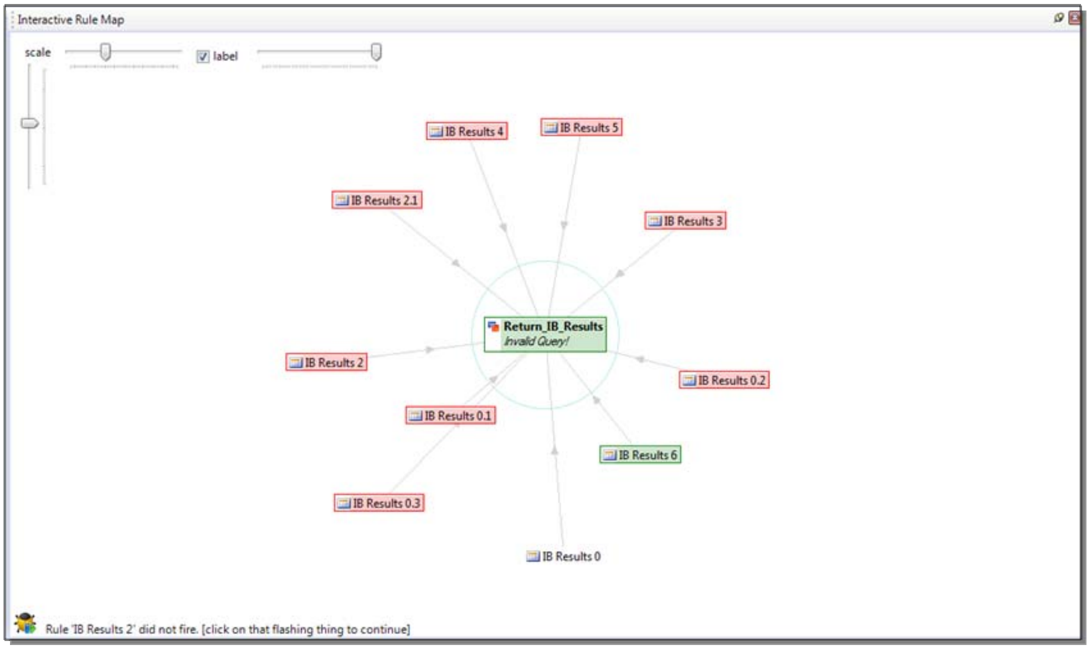
Rule 'IB Results 5' did not fire. [click on that flashing thing to continue]



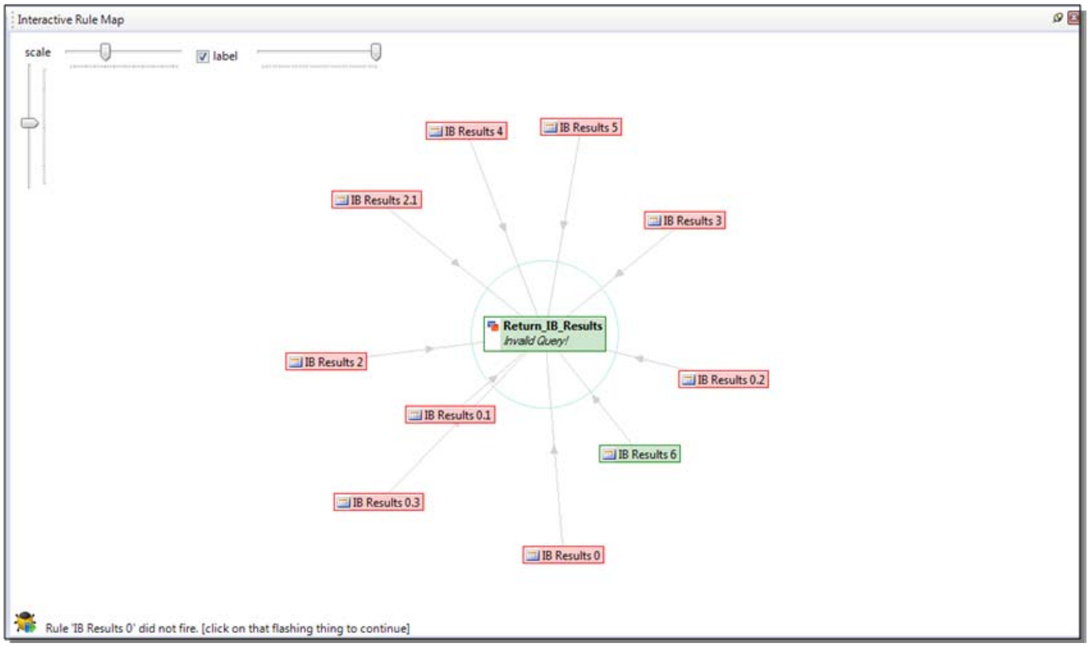
Rule 'IB Results 4' did not fire. [click on that flashing thing to continue]



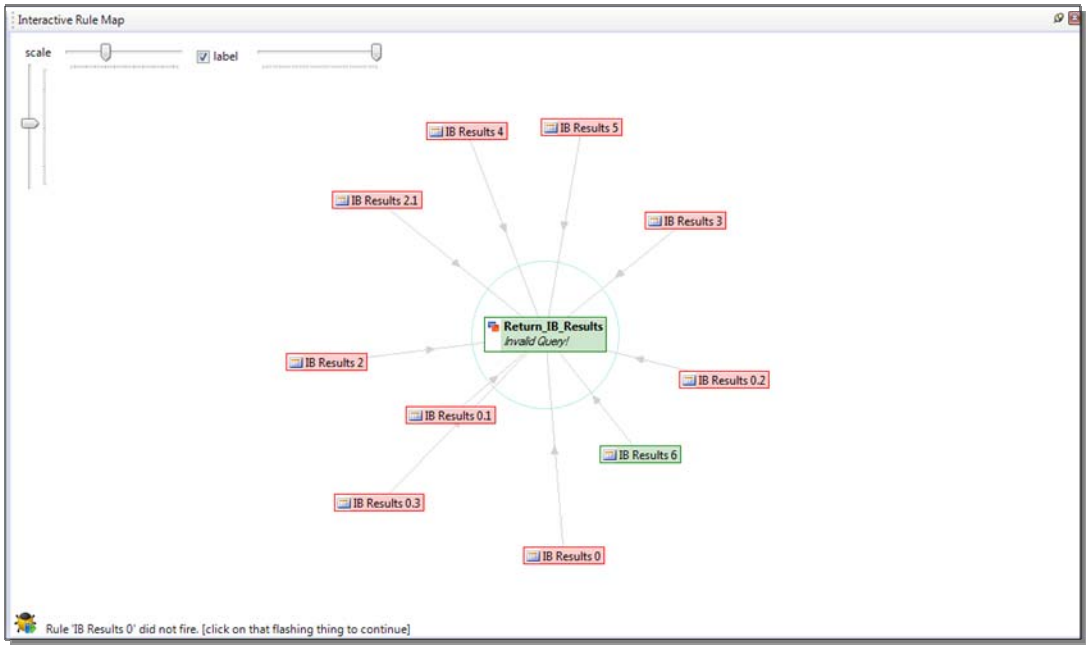
Rule 'IB Results 3' did not fire. [click on that flashing thing to continue]



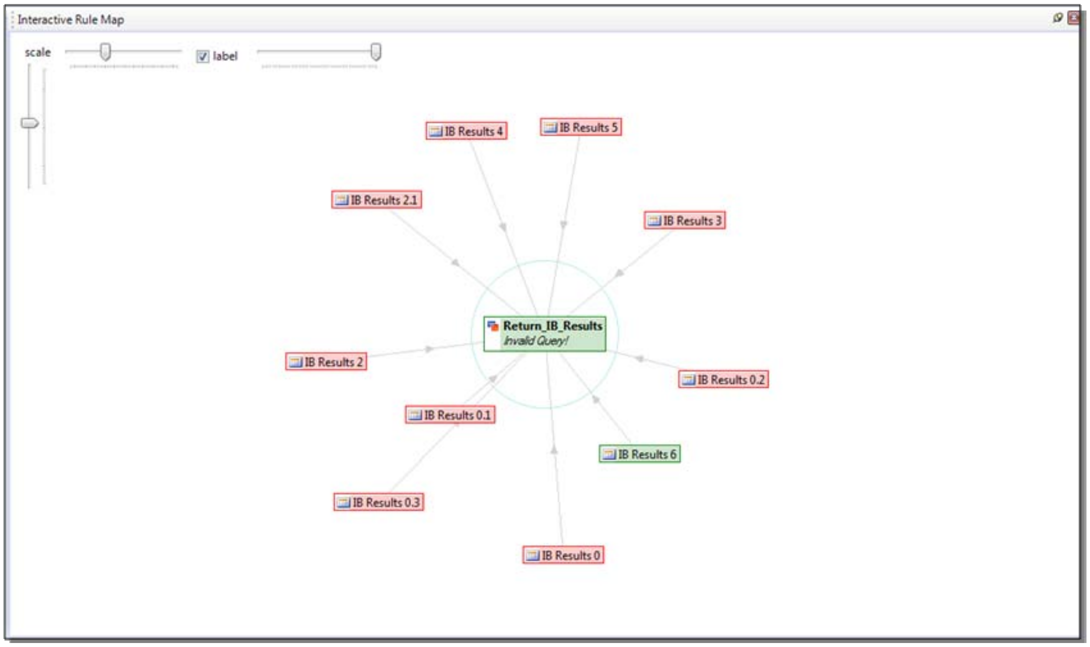
Rule 'IB Results 2' did not fire. [click on that flashing thing to continue]



Rule 'IB Results 0' did not fire. [click on that flashing thing to continue]



Rule 'IB Results 0' did not fire. [click on that flashing thing to continue]



Interactive Rule Map
scale
☑ label
IB Results 4
IB Results 5
IB Results 2.1
IB Results 3
Return_IB_Results
Invalid Query!
IB Results 2
IB Results 0.2
IB Results 0.1
IB Results 6
IB Results 0.3
IB Results 0

LIST OF REFERENCES

[1] M. A. McConnell, M. G. Hayden, R. Mueller, M. L. Maples and R. Fort. Hearing of the Senate Select Committee on Intelligence. Annual Worldwide Threat Assessment. February, 8, 2008.

[2] S. Harris. *CISSP All-in-One Exam Guide, Sixth Edition*. New York: McGraw Hill, 2012.

[3] M. A. Bishop, *Computer Security: Art and Science.* Boston, MA: Addison-Wesley Professional, 2003.

[4] Director of National Intelligence. *Intelligence Community Directive Number 503: Information and Information Systems Security, Risk Management, Certification and Accreditation*. McLean, VA: Office of the Director of National Intelligence, September, 2008.

[5] Department of the Navy, Chief Information Officer. *Maritime Domain Awareness Architecture Management Hub Strategy. 1.0*, Washington, DC: DON, January 2009.

[6] Northrup Grumman Corporation. “Radiant alloy information broker system architecture.” San Diego, CA. PowerPoint presentation, unpublished.

[7] Northrup Grumman Corporation. “Radiant alloy high level concept” San Diego, CA. PowerPoint presentation, unpublished.

[8] W. M. Vanfleet et al., “MILS: Architecture for high assurance embedded computing, STCT Crosstalk,” *Journal of Defense Software Engineering,* vol. 18, pp. 12–16, August, 2005.

[9] U.S. Department of Homeland Security. “National Strategy for Maritime Security: National Plan to Achieve Maritime Domain Awareness.” Washington, DC: DON, October 2005.

[10] The Rule Markup Initiative. The Rule Markup Initiative. August 24, 2008, Available: http://www.ruleml.org.

[11] S. Castano et al. *Database Security*, Boston, MA: Addison-Wesley Publishing Co., 1994.

[12] M. A. Harrison et al. “Protection in Operating Systems.” *Communications of the ACM* 19(8), pp. 461-471. 1976. Available: http://doi.acm.org/10.1145/360303.360333.

[13] I. Sommerville. *Software Engineering (9th ed.)*, Boston, MA: Addison-Wesley, 2010.

[14] J. W. Freeman et al. “A Validated Security Policy Modeling Approach.” *10th Annual Computer Security Applications Conference*, pp. 189-200, Dec. 1994. DOI: 10.1109/*CSAC*.1994.367308.

[15] R. J. Arvay et al. “Using RuleML to Specify Cross-Domain Information Flow Control Policies.” *IEEE International Conference on System of Systems Engineering* 2009, n.p.

[16] C. R. McDaniel, and M. L. Tardy, Role-Based Access Control for Coalition Partners in Maritime Domain Awareness, master’s thesis, Naval Postgraduate School, Monterey, Calif., June 2005.

[17] M. E. Bennett, Defining a Common Intelligence Picture for the United States Coast Guard: A Port Perspective, master’s thesis, Joint Military Intelligence College, Washington, DC, Aug. 2003.

[18] C. Wang and S. Ju. “The New criteria for covert channels auditing,” in *Proceedings from the Fifth Annual IEEE SMC*. 2004. DOI: 10.1109/IAW.2004.1437815.

[19] K. Loepere. “The Covert Channel Limiter Revisited.” *Operating Systems Review* 23(2), pp. 39-44. 1989. Available: http://dblp.uni-trier.de/db/journals/sigops/sigops23.html#Loepere89.

[20] A. B. Shaffer et al. “A Security domain model to assess software for exploitable covert channels.” in *Proceedings of the Third ACM SIGPLAN Workshop on Programming Languages and Analysis for Security*. 2008, Available: http://doi.acm.org/10.1145/1375696.1375703.

[21] T. Jaeger et al. “Managing the risk of covert information flows in virtual machine systems.” in *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*. 2007, Available: http://doi.acm.org/10.1145/1266840.1266853.

[22] N. Nagatou and T. Watanabe, “Run-time detection of covert channels.” in *Proceedings IEEE First International Conference on Availability, Reliability and Security*, September, 2006.

[23] D. E. Bell and L. LaPadula, “Secure Computer Systems: Mathematical Foundations and Model.” MITRE Technical Report 2547, vol. II, pp. M74-75-244, May 1973.

[24] D. E. Denning. “A Lattice model of secure information flow.” *Communications of the ACM* 19(5), pp. 236-243. 1976. Available: http://doi.acm.org/10.1145/360051.360056.

[25] S. Jha et al. “Towards formal verification of role-based access control policies.” *IEEE Transactions on Secure and Dependable Computing* 5(2), 2008. DOI: 10.1109/TDSC.2007.70225.

[26] S. W. Ambler. *The Elements of UML(TM) 2.0 Style*, New York, NY: Cambridge University Press, 2005.

[27] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River, NJ: Prentice Hall, 2005.

[28] T. Erl, *SOA Principles of Service Design*. Boston, MA: Prentice Hall, 2007.

[29] H. Hinton et al. “Security Patterns within a Service-Oriented Architecture.” IBM Websphere, November, 2005. Available: http://searchwebservices.techtarget.com/searchWebServices/downloads/SecuritySOA_(2).pdf.

[30] J. M. Rushby. “Design and verification of secure systems”, in *Proceedings of the Eighth ACM Symposium on Operating Systems Principles*. 1981, p. 12-21, Available: http://doi.acm.org/10.1145/800216.806586.

[31] J. M. Rushby and R. DeLong, Compositional certification, MILS integration protection profile, and Common Criteria authoring environment, Powerpoint Briefing, unpublished.

[32] Green Hills Software, Separation Kernel PowerPoint presentation, unpublished.

[33] S. Blackman, LynuxWorks and the LynxSecure Separation Kernel PowerPoint presentation, unpublished.

[34] SISA Alliance. SISA | Secure Information Sharing Architecture. April, 2008. Available: http://www.sisaalliance.com/.

[35] BAE Systems. STOP OS Trusted Operating System. 2012. Available: http://www.baesystems.com/ProductsServices/bae_prod_csit_xtsstop7.html.

[36] BAE Systems, Next Generation XTS Guards, 2011. Available: http://www.baesystems.com/BAEProd/groups/public/@businesses/@insyte/documents/bae_publication/baes_pdf_next_gen_xts_guards.pdf.

[37] BAE Systems, STOP OS Secure Application Platform, 2011. Available: http://www.baesystems.com/BAEProd/groups/public/documents/bae_publication/bae_pdf_csit_xts_stop7.pdf

[38] BAE Systems, STOP 7 Security Controls, 2010. Available: http://www.baesystems.com/BAEProd/groups/public/documents/bae_publication/bae_eis_pdf_stop7_white_paper.pdf

[39] S. Osborn et al. “Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies”, *Communications. of the ACM*. 3(2), pp. 85-106. 2000. Available: http://doi.acm.org/10.1145/354876.354878.

[40] R. Sandhu et al. “The ARBAC97 Model for Role-Based Administration of Roles.” *ACM Trans. Inf. Syst. Secur.*, 2(1), pp. 105-135. 1999. Available: http://doi.acm.org/10.1145/300830.300839.

[41] R. Sandhu et al. “The NIST model for role-based access control: Towards a unified standard”, in *Proceedings of the Fifth ACM Workshop on Role-Based Access Control*. pp. 47-63, 2000. Available: http://doi.acm.org/10.1145/344287.344301.

[42] D. F. Ferraiolo et al. “Proposed NIST standard for role-based access control”, *ACM Trans. Inf. Syst. Secur.*. 4(3), pp. 224-274. 2001. Available: http://doi.acm.org/10.1145/501978.501980.

[43] L. Chen and J. Crampton. “Inter-domain role mapping and least privilege,” in *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*. 2007, Available: http://doi.acm.org/10.1145/1266840.1266866.

[44] T. Jaeger and J. E. Tidswell. “Practical safety in flexible access control models,” *ACM Trans. Inf. Syst. Secur*. 4(2), pp. 158-190. 2001. Available: http://doi.acm.org/10.1145/501963.501966.

[45] A. Kern et al. “A Meta model for authorisations in application security systems and their integration into RBAC administration,” in *Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies*. 2004, Available: http://doi.acm.org/10.1145/990036.990050.

[46] M. Wilikens et al. “A Context-related authorization and access control method based on RBAC,” in *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies*. 2002, Available: http://doi.acm.org/10.1145/507711.507730.

[47] X. Zhang et al. “An attribute-based access matrix model,” in *Proceedings of the 2005 ACM Symposium on Applied Computing*. 2005, Available: http://doi.acm.org/10.1145/1066677.1066760.

[48] X. Zhang et al. “Safety analysis of usage control authorization models,” in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*. 2006, Available: http://doi.acm.org/10.1145/1128817.1128853.

[49] Committee on National Security Systems, “CNSS Instruction 4009,” *National Information Assurance Glossary*, Washington, DC: 2003.

[50] D. Gray, “Information Support Systems Environment (ISSE) System Engineering and Development for United States Special Operations Command (USSOCOM),” ITT Corporation for AFRL/RIEB, Herndon, VA, Tech. Rep. NA, December, 2010.

[51] P. Mozloom, “Multi Domain Network Management (MDNM) Architecture,” BAE Systems, Rome, New York, Tech. Rep. AFRL-IF-RS-TR-2003-104, May, 2003.

[52] T. Meyers, “Defense Message System Concept of Operations Version I,” Information Assurance Technology Analysis Center, Falls Church, VA, Tech. Rep. IATAC-TAT-02108, 6/1/2003.

[53] M. H. Kang et al. "Design and assurance strategy for the NRL pump,” in *Proceedings of High-Assurance Systems Engineering Workshop,* 1997.

[54] C. E. Irvine et al. “Overview of a high assurance architecture for distributed multilevel security,” in *Proceedings from the Fifth Annual IEEE SMC*. 2004.

[55] Guardian Digital WebTool Firewall HowTo; 4.3 Firewall Rules. Available: http://engardelinux.org/doc/howtos/webtool-firewall-howto/webtool-firewall-howto/x168.shtml.

[56] Cisco Systems. ASA 5500 Adaptive Security Appliance Overview. Available: http://www.cisco.com/en/US/products/ps6120/index.html.

[57] Checkpoint. 61000 Security System. Available: http://www.checkpoint.com/products/61000-appliances/.

[58] McAfee. McAfee Firewall Enterprise. Available: http://www.mcafee.com/us/products/firewall-enterprise.aspx.

[59] Sourcefire. Snort. Available: http://www.snort.org/.

[60] V. Dhankhar et al. "Securing workflows with XACML, RDF and BPEL," in *Proceeedings of the 22nd annual IFIP WG 11.3 conference on Data and Applications Security*, pp. 330-345, 2007.

[61] OASIS, Extensible Access Control Markup Language, Feb 2005. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

[62] S. Ceri et al. "What you always wanted to know about datalog (and never dared to ask)," in *IEEE Transactions Knowledge and Data Engineering,* 1(1), pp. 146-166. 1989. DOI: 10.1109/69.43410.

[63] Accenture. The Trusted Application Layer Interface (TALI): A Software Technology for Organizations Interested in Secure and Discretionary Data Sharing, unpublished.

[64] G. Sindre and A. L. Opdahl, "Eliciting security requirements by misuse cases," in *Technology of Object-Oriented Languages and Systems,* pp. 120-131, 2000.

[65] M. Hartong, R. Goel and D. Wijesekera. Securing positive train control systems, Critical Infrastructure Protection. 2007. DOI: 10.1007/978-0-387-75462-8.

[66] G. B. Meyrick and S. M. McDermott. "Implementing the Defense Message System," *1995 IEEE Military Communications Conference,* 1995. DOI: 10.1109/MILCOM.1995.483500.

[67] P. Case. "Enhancing XML search with XQuery 1.0 and XPath 2.0," in *IBM Systems Journal,* 45(2), pp. 353-360. 2006. DOI: 10.1147/sj.452.0353.

[68] Logic Programming & Intelligent Systems. VDR-Device Tutorial. Available: http://lpis.csd.auth.gr/systems/VDR-Device_Tutorial.htm.

[69] Acumen Business. The Rule Manager. Available: http://www.acumenbusiness.com/products.htm.

[70] E. Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software,* Boston, MA: Addison-Wesley Professional, 1995.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California