NIST

NIST IR
7815

Access Control for SAR Systems

Stephen Quirolgico
Vincent Hu
Tom Karygiannis

Access Control for SAR Systems

Stephen Quirolgico
Vincent Hu
Tom Karygiannis

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8940

July 2011

U.S. Department of Commerce
Rebecca M. Blank, Acting Secretary

National Institute of Standards and Technology
Patrick Gallagher, Under Secretary for Standards and Technology and Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This NIST Interagency Report describes ITL's research, guidance, and outreach efforts in computer security, and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Interagency Report 7815

Disclaimer

This document identifies certain software products, but such identification does not imply recommendation by the US National Institute for Standards and Technology, nor does it imply that the products identified are necessarily the best available for the purpose.

Acknowledgements

The authors wish to thank their colleagues who reviewed drafts of this document. The authors also gratefully acknowledge and appreciate the comments and contributions made by government agencies, private organizations, and individuals in providing direction and assistance in the development of this document.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

In 2009, the National Institute of Standards and Technology on behalf of the Program Manager, Information Sharing Environment (PM-ISE), Office of the Director of National Intelligence (ODNI), initiated the Access Control for SAR Systems (ACSS) project. This project focused on developing a prototype privilege management system used to express and enforce policies for controlling access to Suspicious Activity Report (SAR) data within the law enforcement domain. This work included the design, implementation and integration of distributed software components for rendering policy decisions, storing subject and resource data, and facilitating web-based retrieval of SAR records. In addition, this pilot included work to support the lifecycle of policies used by the privilege management system. This policy lifecycle comprised four phases: (1) identification, (2) encoding, (3) testing, and (4) deployment of policies. The identification phase involved the identification of pertinent laws, statutes and policies for granting or denying access to SAR records. After these laws, statutes and policies were identified, they were encoded into a computer-processable representation during the encoding phase. Encoded laws, statutes and policies were then tested during the testing phase to ensure correctness and accuracy. Finally, during the deployment phase, the encoded laws, statutes and policies were deployed in the developed privilege management system. This general policy lifecycle is shown in Figure 1.

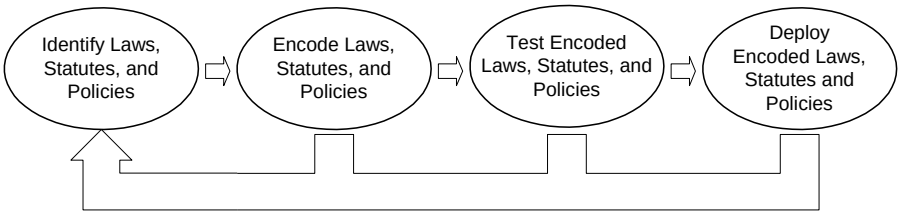

Figure 1. Policy lifecycle phases.

This report details the work conducted for the Access Control for SAR Systems (ACSS) project. We begin by describing the phases of the policy lifecycle and introduce some concepts that are fundamental to the design and implementation of a privilege management system. Next, we describe the design and implementation of the *Policy Evaluation Testbed* (PET) system. The PET system is a prototype privilege management system developed to demonstrate the application of access control policies to SAR records. We then follow with a discussion of the installation and usage of the PET system. This discussion also includes an example use-case scenario that demonstrates the application of policies to SAR records. Next, we describe issues uncovered during the design and implementation of the privilege management system as well during the identification, encoding, testing and deployment of policies. For these issues, we offer general and domain-specific recommendations. Finally, we conclude with a description of future research and development topics aimed at facilitating the development of privilege management systems.

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This document is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), "Securing Agency Information Systems," as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

1.2 Document Scope and Purpose

The purpose of this document is to provide agencies with background information on access control for SAR systems and to assist them in designing and implementing a system for controlling access to SAR data. The document discusses the policy development lifecycle, implementation of a privilege management system for controlling access to SAR data, and issues and recommendations related to the development of access control for SAR systems.

1.3 Audience and Assumptions

This document assumes that the readers have basic knowledge of access control, particularly attribute-based access control. Because of the constantly changing nature of the information technology industry, readers are strongly encouraged to take advantage of other resources (including those listed in this document) for more current and detailed information.

1.4 Definitions

For this report the following terms and definitions apply.

Access Control	A system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system.
Authentication	The act of establishing or confirming someone as authentic.
Authorization	The function of specifying access rights to resources.
PDP	A system entity that evaluates encoded policies and renders an authorization decision.
PEP	A system entity that performs access control by making decision requests and enforcing authorization decisions by the PDP.
Policy	A principle or rule to guide decisions and achieve rational outcome(s).

1.5 Abbreviations

ABAC	Attribute-Based Access Control
ACPT	Access Control Policy Tool
EJB	Enterprise Java Beans
GAF	General Attribute Framework
GFIPM	Global Federated Identity Privilege Management
GWT	Google Web Toolkit
HSIN	Homeland Security Information Network
ISE	Information Sharing Environment
NIEM	National Information Exchange Model
NIST	National Institute of Standards and Technology
ODNI	Office of the Director of National Intelligence
OWL	Web Ontology Language
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PET	Policy Evaluation Testbed
PII	Personally Identifiable Information
PIP	Policy Information Point
PM-ISE	Program Manager – Information Sharing Environment
RDF	Resource Description Framework
RPC	Remote Procedure Call
SAR	Suspicious Activity Report
SPARQL	SPARQL Protocol and RDF Query Language
SLT	State/Local/Tribal
SSL	Secure Sockets Layer
TLS	Transport Layer Security
XACML	eXtensible Access Control Markup Language

2 POLICY LIFECYCLE

2.1 Policy Identification

The first phase of the policy lifecycle involves the identification of laws, statutes and policies pertinent to granting or denying access to domain resources. The identification of laws, statutes and policies pertinent to granting or denying access to domain resources requires not only intimate knowledge of pertinent federal, state, local, tribal, and domain-specific laws, statutes and policies, but also of domain resources, it's users and actions that users may invoke on those resources. In general, access control policies are motivated by laws, statutes, and policies that ensure the privacy of sensitive information including classified data and personally identifiable information (PII).

After pertinent laws, statutes and policies have been identified, work must be conducted to map these laws, statutes and policies to domain-specific policies. The first step in this process entails the mapping of general laws, statutes and policies into *Attribute-Based Access Control* (ABAC) [Blaze, Brown, Pimlott] rules. An ABAC rule is a triple $\langle SA, A, RA \rangle$ where *SA* represents a set of domain subject (i.e., user or process) attributes, *RA* represents a set of domain resource attributes, and *A* represents a set of actions that users may invoke on resources. The set SA is defined by a subject attribute data model that defines subject attributes (and their interrelationships) required to render a policy decision (i.e., permit or deny access to a resource). Such attributes may include, for example, the role, rank and operating unit of an employee. The set RA is defined by a resource attribute data model that defines resource attributes (and their interrelationships) required to render a policy decision. Such attributes may include, for example, the fields of a resource database record or the clearance level of the resource (i.e., public, secret, or top secret). Set A defines a set of actions that a user having a subset of attributes in SA may invoke on a resource having a subset of attributes in RA. Actions may include "READ", "WRITE", or "EXECUTE" operations. Other attribute data models may also exist including environment and obligation attribute data models. Figure 2 shows an example mapping of laws, statutes and policies to an ABAC rule for a law enforcement domain. Here, we use the Global Federated Identity Privilege Management (GFIPM) [GFIPM] specification to represent the subject attribute data model and the NIEM SAR [NIEM-a, NIEM-b] specification to represent the resource attribute data model. In addition, we use the set {"R","W","E"} (i.e., "READ","WRITE","EXECUTE") to represent the set of possible actions.

Before a policy can be derived from an ABAC rule, a new rule must be defined that specifies subsets of the ABAC rule. These subsets of attributes reflect the attributes used for a specific policy. That is, we let $\langle SA', A', RA' \rangle$ represent the triple of ABAC rules where $SA' \subseteq SA$ represents a set of subject attributes, $A' \subseteq A$ represents a set of action attributes and $RA' \subseteq RA$ represents a set of resource attributes for some specific policy.

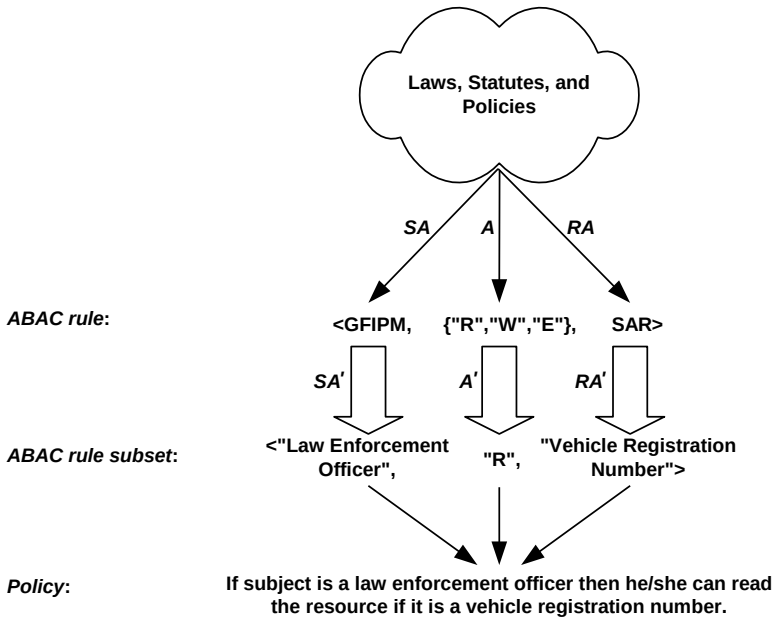

Figure 2. Example SAR policy derivation.

For example, a policy may only require the subject attribute "Law enforcement officer" from the set of attributes defined in the GFIPM data model, the "R" (Read) action and the "Vehicle Registration Number" resource attribute from the set of attributes defined in the SAR data model. This ABAC rule subset is shown in Figure 2.

We define an *ABAC policy* as a set of ABAC rules. In the current example, we assume that the attributes of ABAC rules have a Boolean data type with true and false values. Thus, the policy reads: If it is true that the subject is a law enforcement officer and it is true that the resource to be accessed is a vehicle registration number and it is true that the access action is read, then the permission is true.

2.2 Policy Encoding

After all policies have been defined, they must be encoded in a computer-processable representation such as the eXtensible Access Control Markup Language (XACML)[OASIS-a], Resource Description Framework (RDF) [W3C-a] or Web Ontology Language (OWL) [W3C-b]. Often, such representations are coupled to components, systems or platforms for supporting the creation, maintenance, processing, and sharing of policies. For example, policies encoded in an XACML document will require a system for supporting the maintenance, processing and sharing of XACML-related documents. The selection of an encoding representation affects many aspects of a privilege management system including (1) the ability to support certain types of policies,

(2) the way in which policy decisions are requested and responded, (3) the components and infrastructure required for supporting the encoded representation, (5) the ease of creating and maintaining encoded policies, and (6) the interoperability with other systems.

2.3 Policy Testing

After policies have been encoded, they must be tested to ensure that they are complete and correct. This is particularly true when multiple policies are considered when rendering the final decision since discrepancies or conflicts may exist. Identifying discrepancies or conflicts between policy specifications and their intended function is crucial to the proper operation of a privilege management system. As a result, policy specifications must undergo rigorous verification and validation to ensure that the policy specifications truly encapsulate the desires of the policy authors.

2.4 Policy Deployment

The deployment of encoded policies involves making them available to the system component that renders policy decisions. However, other components of a privilege management system may also need to be tailored to support encoded policies. To better understand how components of a privilege management system must be designed and implemented to support encoded policies, it is necessary to understand the design and implementation of a privilege management system. In Section 3, we describe the design and implementation of the Policy Evaluation Testbed privilege management system.

3 PET DESIGN AND IMPLEMENTATION

The *Privilege Evaluation Testbed* (PET) system is a system aimed at demonstrating the application of policies for controlling access to Suspicious Activity Report (SAR) data. This section defines the design and implementation of PET v1.0, the baseline version of PET that embodies the minimal components and mechanisms required for applying basic access control to SAR records.

3.1 Functional Scope

The scope of the design and architecture of PET v1.0 is driven by a set of functional requirements. These functional requirements include supporting:

- remote, web-based access to SAR records
- fundamental authentication
- fundamental querying of SAR records
- fundamental access control policies for granting or denying access to SAR records
- the use of basic subject, resource and environment attributes
- fundamental infrastructure security
- a subset of SAR-related access control policies highlighting specific Federal, State, Local, and Tribal laws, statutes and policies

Note that this version of PET is a prototype system; it is not a full and comprehensive production system. Thus, this version of PET does not provide features that might be found in such a system including a complete set of SAR-related policies.

3.2 Architectural Overview

PET is a distributed system that follows the non-normative XACML data-flow model [OASIS-a] as shown in Figure 3. The *Policy Decision Point* (PDP) is the system entity that evaluates encoded policies and renders an authorization decision. Encoded policies are made available to the PDP by a *Policy Administration Point* (PAP). The PDP interacts with the rest of the access control system through the system's *Context Handler*. The Context Handler is the system entity that converts decision requests in the native request format to the XACML canonical form and converts authorization decisions in the XACML canonical form to the native response format. The context handler receives decision requests from the *Policy Enforcement Point* (PEP). The PEP is the system entity that performs access control by making decision requests and enforcing authorization decisions by the PDP. The PEP receives access requests from an *Access Requester*. An Access Requester may be any entity that requires access to a resource including a machine process requiring access to a file or a user requiring access to a database record. The *Policy Information Point* (PIP) provides the interface to access subject, resource and

other attributes. A more detailed description of the XACML data-flow model is found in [OASIS-a].

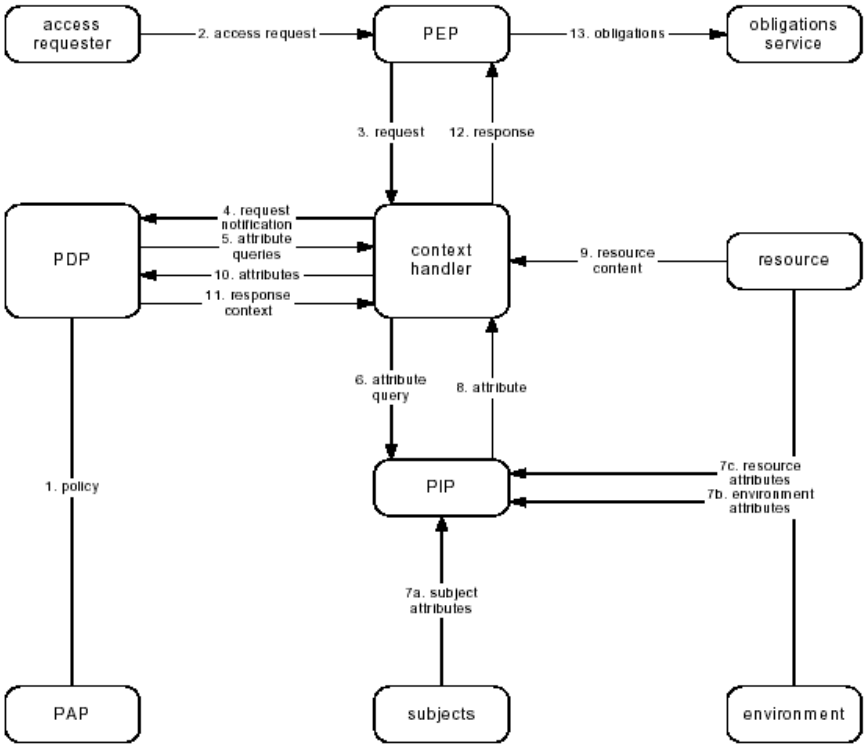

Figure 3. XACML Data-Flow Model [OASIS-a].

The PET system embodies a simplification of the XACML data-flow model architecture. For instance, PET combines the functionality of the PEP and Context Handler into a single PEP entity. In addition, certain components of the XACML data-flow model are omitted including the PAP (since the implemented PDP provides direct access to the system's policies) and obligation services. The simplified PET architecture is shown in Figure 4.

3.3 Concept of Operations

The operation of the PET begins with a user authentication process. During the user authentication process, a user submits a login ID and password to the PEP via the PET (login) web interface. After receiving a user ID and password, the PEP attempts to authenticate the user by matching the received user ID and password against entries in the Subject (User) database. If a match exists, the PEP will (1) authorize the user for access to the PET system, (2) create a new user session, and (3) return the main PET page. If a match does not exist, however, the PEP returns an error to the user. The PET user authentication process is shown in Figure 5.

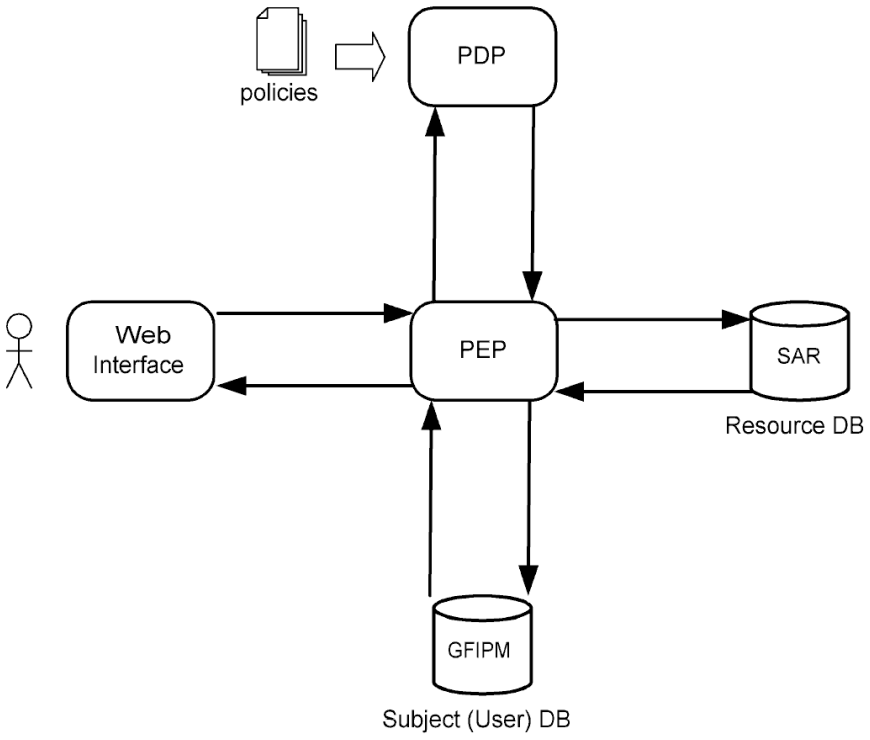

Figure 4. PET architecture.

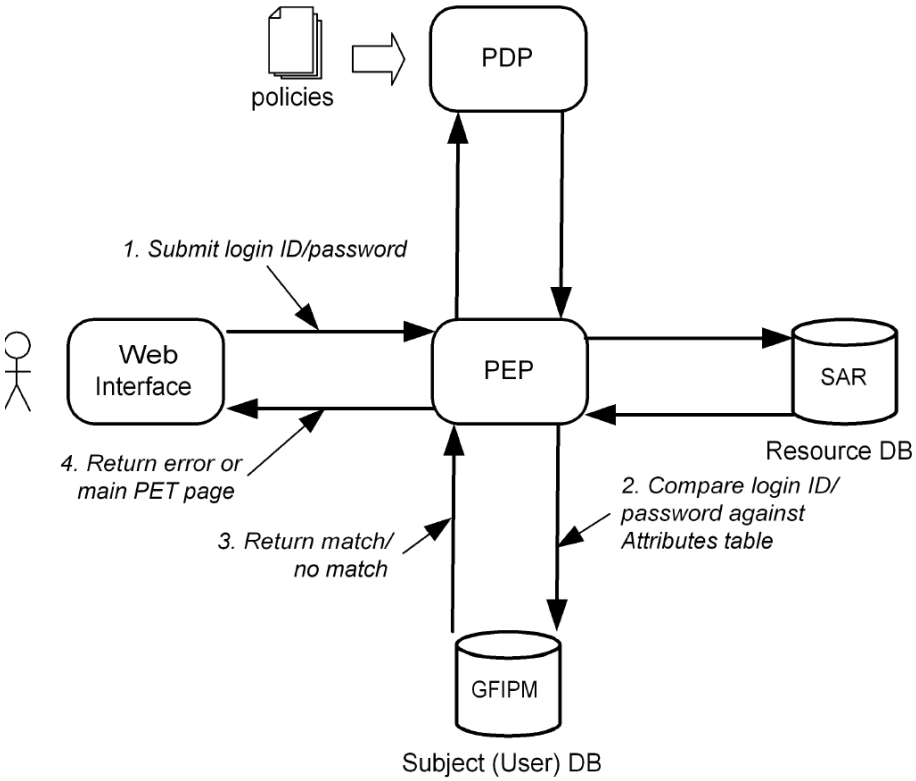

Figure 5. PET user authentication.

After being authorized by the PEP, the user may then query the system for SAR records. When querying for SAR records, the user will enter the query in the PET web interface which, in turn, is submitted to the PEP. The PEP then proceeds to submit one or more XACML Requests to the PDP to determine what level of access, if any, the user has to SAR records. If the user is not authorized to access any SAR records, an error is returned to the user. If the user is authorized to access a SAR record, the PDP will then identify the types of SAR records the user is authorized to access: Full, Summary ISE, or Summary SLT SAR records. A Full SAR record is a SAR record containing values for all fields. Both a Summary ISE (Information Sharing Environment) record and a Summary SLT (State/Local/Tribal) record is a SAR record containing values for only a subset of fields. For these records, fields are omitted based on the implemented policies. Such policies, for example, may omit values for SAR fields that describe personally identifiable information (PII).

Given the policies maintained by the PDP, the PDP will render a policy decision whether the user is authorized to access Full, Summary ISE, Summary SLT, or no SAR records. If the user is not authorized to access any SAR records, the PDP will deny the request and the PEP will return an error to the user. However, if the user is authorized to access a Full, Summary ISE, or Summary SLT SAR record, the PDP will permit the request and inform the PEP of this decision. The PEP will then query the SAR database for matching SAR records, filter fields from these records if necessary (to derive Summary ISE or Summary SLT records) and return the records to the user. The SAR record query process is shown in Figure 6.

3.4 PET Components

The implementation of PET involves the development, configuration, and integration of several components, technologies and standards. The PET implementation architecture is shown in Figure 7.

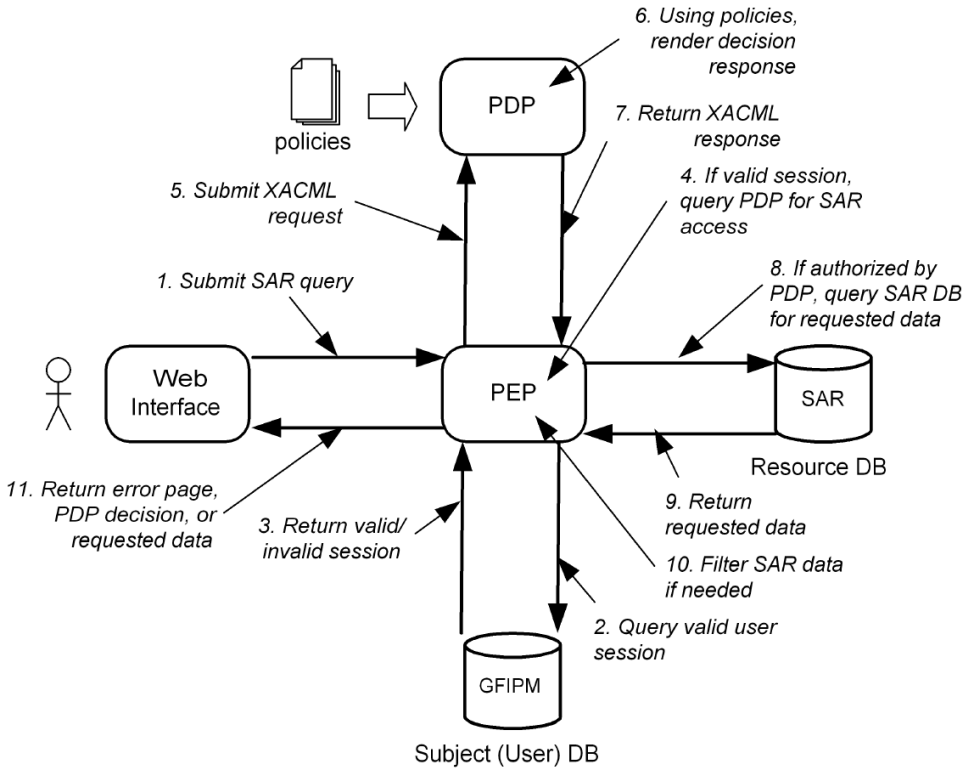

Figure 6. Querying SAR records.

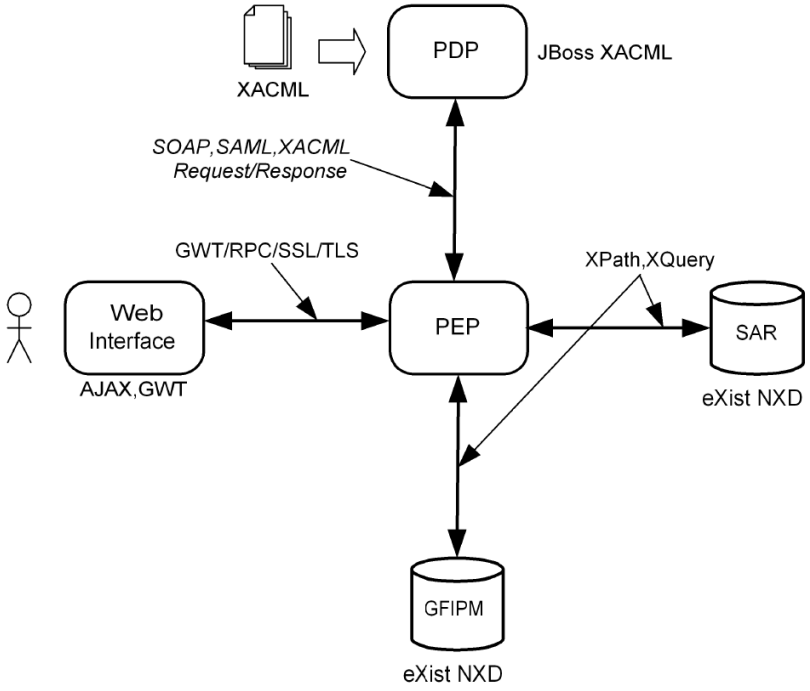

Figure 7. PET implementation.

3.4.1 Web Interface

The PET web interface supports user login, querying and displaying of SAR records, and logout functionality. These functions are communicated to the PEP via the Google Web Toolkit (GWT) [GWT] Remote Procedure Call (RPC) mechanism allowing the PEP to push data to the interface. With respect to security, secure communication via Secure Sockets Layer (SSL) and Transport Layer Security (TLS) is implemented between the PET user interface and the PEP host server.

3.4.1.1 Login Interface

The PET login interface contains fields for user ID and password as well as a button for submitting user input data to the PEP. Here, an RPC is invoked to submit login information to the PEP. The PET login interface is shown in Figure 8.

The PET query interface provides mechanisms for logging out of the PET system, querying SAR records, and viewing query results.[1] The PET user interface page, which is generated dynamically by the PEP, contains forms for data input. The SAR query user interface page provides "Google-like" search capabilities by allowing queries to be entered as keywords. For two or more keywords, a conjunctive operator is used. Figure 9 shows a query with the keywords "Springfield Nuclear Power Plant" from a database maintained by State C.[2]

3.4.1.2 Query Interface

The PET query interface page displays retrieved SAR records in their original XML format. Note that the query interface informs the user that certain policy restrictions for accessing SAR records may apply. This is achieved through a status bar indicator as well as the blocking of certain SAR record field values (denoted in red font). Figure 10 shows a retrieved SAR record.

[1] Additional functionality including administering SAR data, viewing access control policies, and viewing data from user tables is not required for PET v1.0 but may be added in future versions.

[2] To generalize the states used in demonstrations, PET represents states using the letters "A", "B", etc. rather than actual state names.

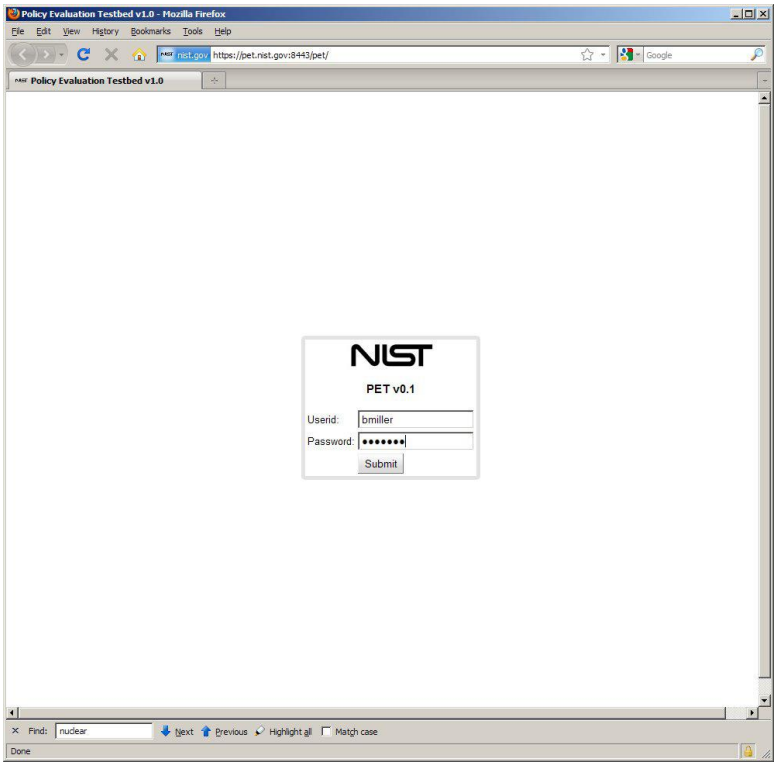

Figure 8. PET login interface.

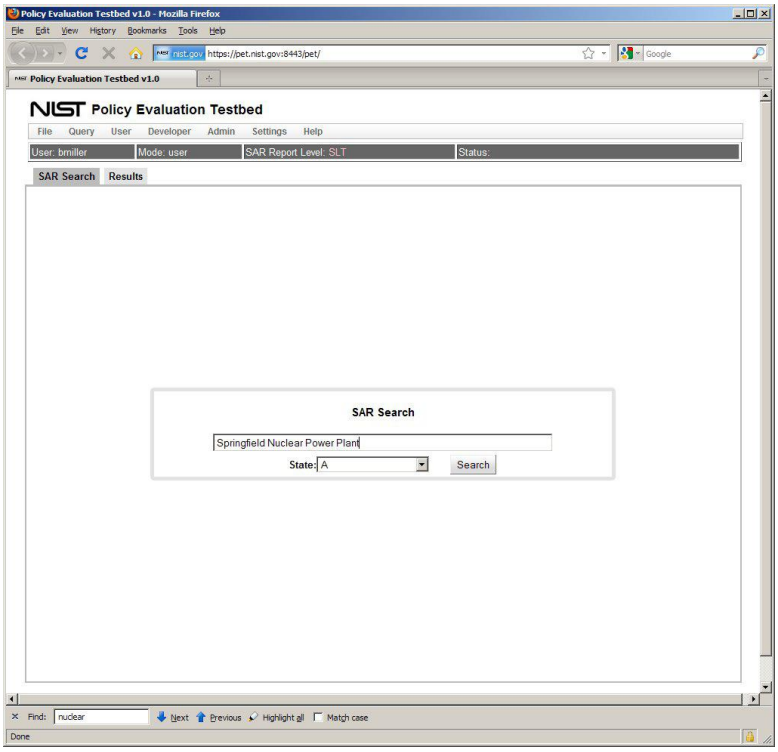

Figure 9. PET query interface.

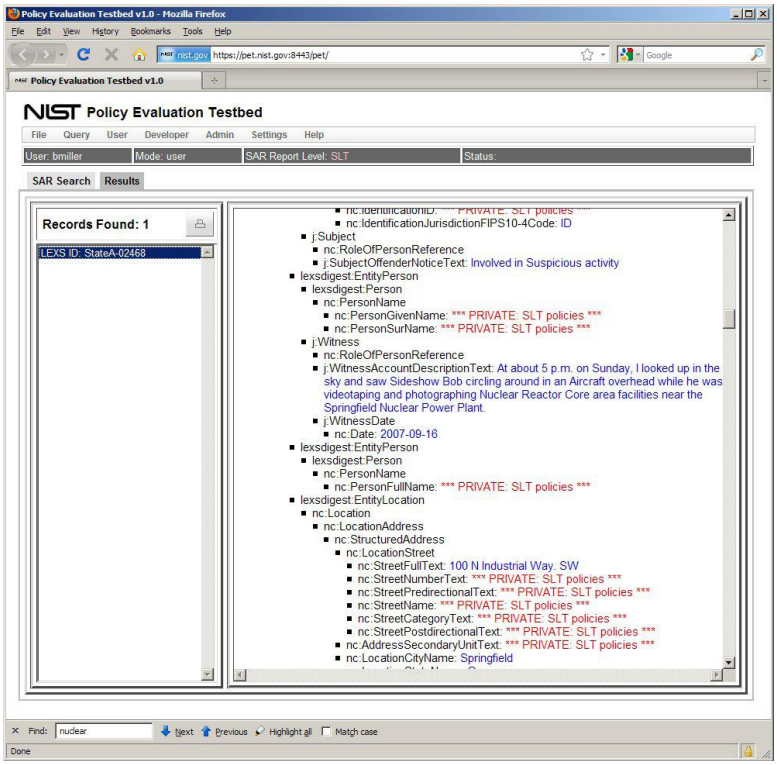

Figure 10. Retrieved SAR record.

3.4.2 Policy Enforcement Point

In the PET system, PEP and Context Handler functionality are combined into a single PEP. Thus, the PEP handles core functions including (1) login authentication, (2) session handling, (3) query handling for SAR records, and (4) result handling. These functions involve interaction with the PDP as well as with remote attribute services or databases. The PEP is implemented as a Java servlet running on an Apache Tomcat servlet container. Here, GWT was used to create RPC functions for the PEP including login authentication and query handling.

3.4.2.1 Login Authentication

The PEP handles login authentication by matching received user ID and passwords against the attributes table in the subject (user) database.[3] This requires the PEP to construct a query to the subject database. If a match exists, the PEP will (1) authorize the user for access to the PET system, (2) create a new user session, and (3) return the main

[3] PET v1.0 assumes that the attribute table in the subject database is populated a priori. PET v1.0 does not provide support for modifying data in the subject database.

PET page to the user. If the PEP cannot match a received user ID and password against the subject database, the PEP returns an error.

3.4.2.2 Session Handler

The PET system uses GWT servlet sessions to maintain state of user activity. A PET session begins once a user is authenticated to the system during login and remains active until the user logs off of the system. PET does not use cookies, Enterprise Java Bean (EJB) Sessions or other methods to maintain sessions.

3.4.2.3 Query Handler

The primary function of the PEP is to restrict unauthorized access to SAR records. When the PEP receives a user query for SAR records, it creates a series of queries to the PDP to determine whether the user is authorized to access Full, Summary ISE, or Summary SLT SAR records. This involves having the PEP construct an XACML Request for each query that represents an ABAC policy comprising a set of subject attribute values, an action, and a set of resource attribute values. To acquire the attribute values needed for the XACML Request, the PEP queries the subject, environment and resource attribute databases. To achieve this, the PEP must consult the policies used by the PDP in order to acquire the proper attributes. Note that in this version of PET, the PEP is responsible for ensuring that attributes needed by the PDP to render a policy decision are supplied in an XACML Request rather than relying on the PDP to acquire these attributes itself.[4]

3.4.2.4 Result Handler

After the PEP submits an XACML Request to the PDP, the PDP will respond with one of four possible XACML Responses: (1) `Permit`, (2) `Deny`, (3) `Indeterminate`, or (4) `NotApplicable`. If the PEP receives a Deny response from the PDP, the PEP will return an error to the user indicating that access to the requested SAR record has been denied. If the PEP receives an `Indeterminate` response from the PDP, the PEP will return an error page to the user indicating that an error occurred or that some value was missing. If the PEP receives a `NotApplicable` response from the PDP, the PEP will return an error to the user indicating that the there is no policy that matches the requested target. If, however, the PDP authorizes access to a Full, Summary ISE or Summary SLT SAR record, the PEP will then query the resource attribute database for matching SAR records. If all keywords in a user query match strings in any field of a SAR record, the PEP will acquire the SAR record and prepare to send it to the user. If the user is authorized to access Full SAR records, the PEP will simply forward the record unmodified to the user. However, if the user is authorized to access only Summary ISE or Summary SLT SAR records, the PEP will remove the value of any field defined in the Summary ISE or Summary SLT private fields lists. These lists are defined in the NIEM SAR specification [NIEM-b]. After a SAR record has been filtered by removing the

[4] At the time of PET v1.0 development, the JBoss XACML service did not provide database hooks to directly acquire required attributes from a native XML database (where PET subject, environment and resource attributes are stored).

values for private fields, the PEP inserts a new String value citing the related policy name. For example, a user with access to only Summary SLT SAR records will show the value " `*** PRIVATE: SLT policies ***` " for the Summary SLT private SAR field `nc:PersonFullName`.

3.4.3 Policy Decision Point

PET uses JBoss XACML [Saldhana] as its PDP implementation. JBoss XACML is an opensource, servlet-based PDP implementation. In PET, the JBoss XACML PDP was treated as a "black-box" that accepted XACML Requests and responded with XACML Responses. In this pilot, no modification of the JBoss XACML source code was conducted.

3.4.4 Resource Attribute Database

PET uses the native XML database (NXD) eXist-db [Meier] to store SAR records that conform to the NIEM SAR [NIEM-b] specification. The eXist-db database supports storing of native XML documents as well as supporting XPath [W3C-e] and XQuery [W3C-f] query languages. The eXist-db API also supports XML RPC for calling remote database procedures. For this pilot, example SAR records were developed to highlight features of the PET system within the context of specific use-case scenarios.

3.4.5 Subject Attribute Database

PET uses the native XML database (NXD) eXist-db [Meier] to store GFIPM records that conform to the GFIPM [GFIPM] specification. For this pilot, example GFIPM records were developed to highlight features of the PET system within the context of specific use-case scenarios.

3.4.6 Access Control Policies

The policies (including the subject and resource attributes and policy actions) used by PET were created to demonstrate use-cases highlighting specific federal, state and local laws, statutes and policies. One such use case is described in Section 5 Use-Case Scenario.

3.5 Communication Mechanisms

The distributed architecture of PET requires the implementation of communication protocols and messaging formats between remote components.

3.5.1 User Interface and PEP

The PET web interface communicates with the PEP via GWT RPCs using SSL/TLS for securing the communication channel. No particular messaging format is used between the user interface and the PEP.

3.5.2 PEP and PDP

With respect to communication with the PDP, XACML Request and Response messages are communicated between the PEP and PDP using the Simple Object Access Protocol (SOAP) [W3C-d] and the Security Assertion Markup Language (SAML)[OASIS-b].

3.5.3 PEP and Attribute Databases

With respect to communication with the subject (GFIPM) and resource (SAR) attribute databases, the PEP uses XPath and XQuery messages over XML RPC to communicate with the eXist-db API.

3.6 Security

The PET system supports SSL and TLS for securing communication channels between the user interface and the PEP host server. In addition, the PEP supports enforcement of access control policies rendered by the PDP. No additional security mechanisms are supported.

3.7 Standards and Open-Source Platforms

To facilitate portability, adoption and cost, the PET system uses only open-source components, platforms and technologies as well as several standards.[5] These are shown in Table 1.

Table 1. Referenced standards and open source components.

Name	Standard	Open Source Component
XACML	X	X
SOAP	X	X
SAML	X	X
JBoss XACML		X
eXist		X
SAR	X	
GFIPM	X	

[5] SAR and GFIPM may be described as emerging standards.

4 PET INSTALLATION AND USER'S GUIDE

Because of the distributed nature of PET, each component must be downloaded, configured, and run to support PET operation. Table 2 shows the package or system used to implement each PET component.

Table 2. Systems for supporting PET components.

PET Component	System or Package
Policy Decision Point (PDP)	JBoss Identity Stack (Java Servlet)
Web Interface	HTML, Javascript, AJAX
Policy Enforcement Point (PEP)	NIST Developed Java Servlet
(SAR) Resource Attribute Database	eXist XML Database
(GFIPM) Subject Attribute Database	MySQL

Installation of a PET requires the following:

- Windows, Macintosh, or UNIX platform
- Java 1.6 or greater
- Apache Tomcat or other servlet container
- eXist-db

In the following, we describe the installation of PET v1.0. All PET installation files are found in the `pet_v1.0.zip` distribution. Unzip `pet_v1.0.zip` into a directory on your system. In the following, we let `$PET` denote the root directory of the unzipped `pet_v1.0.zip` distribution. The `$PET` directory contains the following file and subdirectories:

4.1 Quick Start

This section describes the installation and configuration of the PET v1.0 system for supporting the use-case scenario as described in Section 5.

4.1.1 Subject and Resource Attribute Databases

Both the GFIPM subject and SAR resource attribute databases are deployed using eXist db.

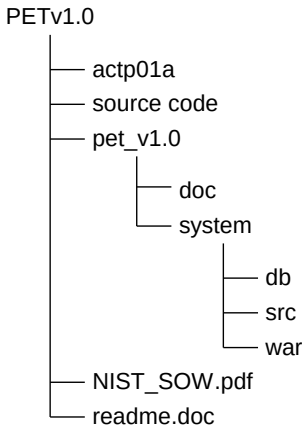

Figure 11. PET directory.

4.1.1.1 Install

Download and install eXist v1.4 or later from http://exist-db.org/. Since eXist is Java-based, it is necessary to ensure that a Java SDK is already installed on your system.

4.1.1.2 Configure

eXist can be deployed in one of three modes (servlet, standalone, and application-embedded). Install xExist in standalone mode. Configure eXist for port 8088.

To start the eXist server, go to `$EXIST/bin/server.bat` (Windows) to start the server. Then, start the eXist client `$EXIST/bin/client.bat` and point its URI to server `xmldb:exist://localhost:8088/xmlrpc`. To populate eXist with both GFIPM attributes and SAR records, select the "Restore files from backup" icon and select the file `eXist-backup.zip` from `$PET\pet1.0a\system\db`. This file contains predefined GFIPM and SAR records used to support the use-case scenario described in Section 5.

4.1.1.3 Test

To test the configuration of the eXist-db database on the local host, point your browser to http://localhost:8088/ and ensure that the GFIPM and SAR collections are visible.

4.1.2 Tomcat

The PET system is a web-based application implemented as a Java servlet. For PET v1.0, we use the Apache Tomcat servlet engine/container. In the following, `$CATALINA` denotes the root directory of the Tomcat installation.

4.1.2.1 Install

Download and install Apache Tomcat 6.x or later.

4.1.2.2 Configure

Before configuring, shutdown Apache Tomcat. Configure Apache Tomcat to use SSL/TLS on port 8443. This will require the creation of a public key certificate. For this pilot, we generated a public key certificate using the Java `keytool` application. Next, install the JBoss XACML PDP by copying the files and folders in `$PET\pet1.0a\system\pdp\jboss-pdp` to the `$CATALINA\webapps` directory. To install the PET system, copy the files and folders in `$PET\pet1.0a\system\war` to the `$CATALINA\webapps` directory and then rename the `\war` directory to `\pet` within the `$CATALINA\webapps` directory.

4.1.2.3 Test

To test PET is operating on the local host, point your browser to http://localhost:8443/pet. Here you will see the PET login page.

4.2 Source-Code Modification and Installation

The PET distribution is provided as an Eclipse/GWT project. Here, the PET Java source code is found in `$PET/pet1.0a/system/src`. Jar files required for PET are located in $`PET/pet1.0a/system/war/WEB-INF/lib`.

5 USE-CASE EXAMPLE

This section describes details of a SAR use-case scenario. In the following, we label specific states by letters "A", "B", and "C" rather than actual state names.

5.1 Suspicious Activity

State A receives a SAR with PII regarding a name, Bob Smith, and license tag for an individual who has been seen with what appear to be small Ziploc packets of white powder near crowded tourist locations (and who travels to and away from the locations via car). The State A police officer who reported the SAR highlights that Bob Smith, rather than acting in the manner expected of a drug dealer, seems to be merely loitering in areas with high foot traffic and keeping the ziploc bags next to a cup of coffee.

5.2 State SAR Data

This scenario involves SAR data separately maintained by three states: State A, State B, and State C. For simplicity, each state will only have a single SAR record pertaining to the suspicious incident scenario in its database. Table 3 describes the SAR attribute values associated with this incident scenario maintained by each state.

Table 3. SAR field values for States A, B, and C.

State	SAR attribute/value	SAR XPath
A	Name/"Bob Smith"	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityPerson[1]/lexsdigest:Person/nc:PersonName/nc:PersonFullName
	License Tag / "123456"	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityVehicle/nc:Vehicle/nc:ConveyanceRegistrationPlateIdentification/nc:IdentificationID
	Suspicious Behavior/ "loitering in high foot-traffic areas with ziploc bags of white powder"	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityActivity[1]/nc:Activity/nc:ActivityDescriptionText
B	License Tag / "123456"	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityVehicle/nc:Vehicle/nc:ConveyanceRegistrationPlateIdentification/nc:IdentificationID

	Suspicious Behavior / “paying cash forlarge amount of common chemical predicate for a chemical Agent”	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityActivity[1]/nc:Activity/nc:ActivityDescriptionText
C	Name / “Bob Smith”	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityPerson[1]/lexsdigest:Person/nc:PersonName/nc:PersonFullName
	Address / “789 Cook St.”	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityLocation[1]/nc:Location/nc:LocationAddress/nc:StructuredAddress/nc:LocationStreet/nc:StreetFullText
	Co-habitants / “Bart Simpson” and “Herman Munster”	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityPerson[1]/lexsdigest:Person/nc:PersonName/nc:PersonFullName
	Suspicious Behavior / “refused medical services for visible burns to his hands”	/lexspd:doPublish/lexs:PublishMessageContainer/lexs:PublishMessage/lexs:DataItemPackage/lexs:Digest/lexsdigest:EntityActivity[1]/nc:Activity/nc:ActivityDescriptionText

5.3 State Access Control Policies

Each state has separate policies for controlling access to its SAR data. Here, we define two-levels of access control for all states: *general authorization* and *record-level authorization*. General authorization refers to a state’s policy that allows access to the state’s SAR repository while a record-level authorization refers to a state’s policy that allows access to full or summary SAR records. For each state, when a user queries SAR records, the state’s SAR system will first determine the general authorization for the user (i.e., whether the user can access the state’s SAR repository). If the user is permitted to access the state’s SAR repository, then the state determines the record-level authorization for the user (i.e., whether the user can access full or summary SAR records). Table 4 describes general authorization policies. Note that we assume that user is already authenticated to the PET system.

Table 4. State general authorization policies.

State	General authorization policy
A	User must have "mere" suspicion to access SAR repository.
B	User must legitimate law enforcement and public protection purposes and only for the performance of official duties in accordance with law
C	User must have "law enforcement or criminal investigative authorities" and "reasonable suspicion for collection or dissemination".

All states have the same policy regarding record-level authorization as shown in Table 5.

Table 5. State record-level authorization policies.

Record Level	Policy
Full SAR Record	User must have (1) remote access and assurance level of 3 or above, or (2) local access and assurance level of 2 or above
Summary	Otherwise

Note that summary SAR records are filtered based on the SLT Private Fields Specification contained in [NIEM-b].

5.4 Users

This demonstration has two users: Bob Richards and Bruce Miller. The subject attributes for these users (subjects) is shown in Table 6.

Table 6. Subject (user) attributes.

Name	Home State	SwornLEO	Public Health/CDC	Assurance Level	Access Mode
Bob Richards	A	F	T	3	Remote
Bruce Miller	A	T	F	2	Remote

5.5 PET Use-Cases

The following describes two use-case examples: Summary SLT and Full SAR Record. In the former, the user is authorized to access only Summary SLT SAR records. In the latter, the user is authorized to access Full SAR records. These use-cases can be run online at https://pet.nist.gov:8443/pet/. When running these use-cases online, you will be asked if you trust the site or will accept a NIST certificate. Accept the certificate to continue to the PET system.

5.5.1 Summary SLT Example

In the first scenario, Bruce Miller (id: `bmiller`, password: `bmiller`) logs into the PET system as shown in Figure 12.

After Miller logs into the system, he may query for SAR records from different states. In this case, he queries for "`Springfield Nuclear Power Plant`" from State C's SAR repository as shown in Figure 13.

Given State C's requirement for general authorization, Miller is prompted to comply with State C's policies as shown in Figure 14.

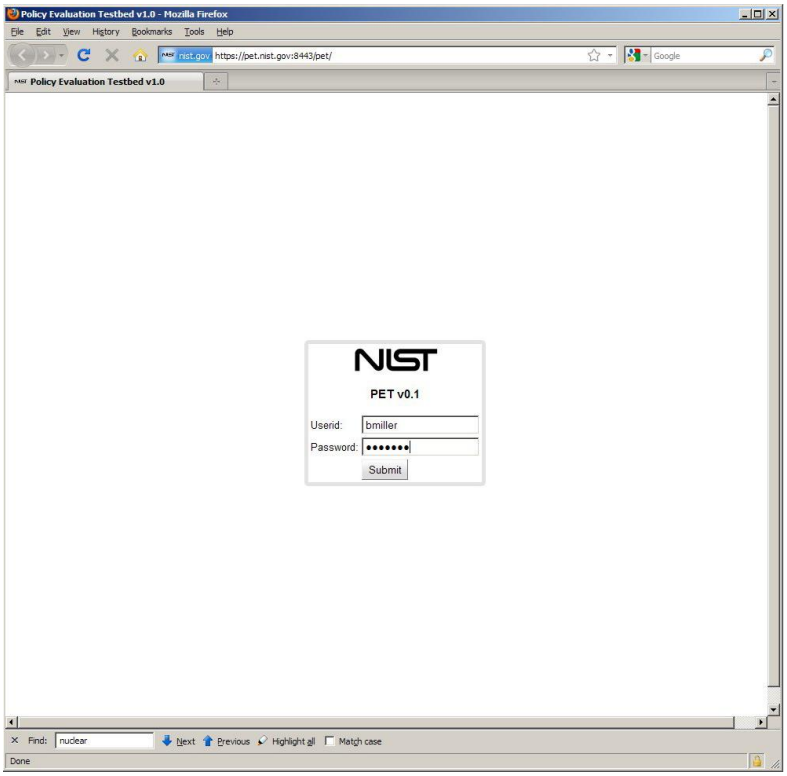

Figure 12. Subject (user) attributes.

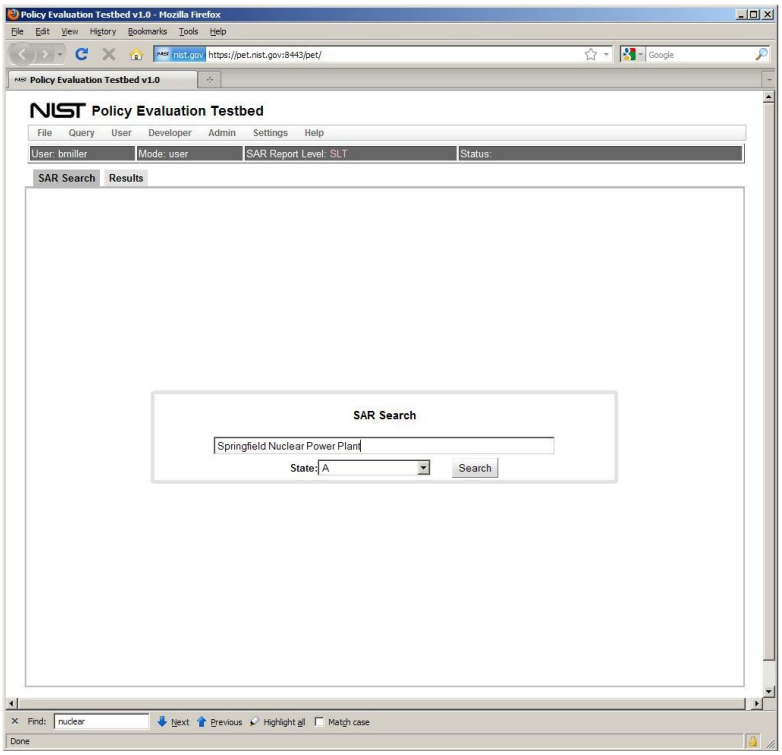

Figure 13. Querying SAR records.

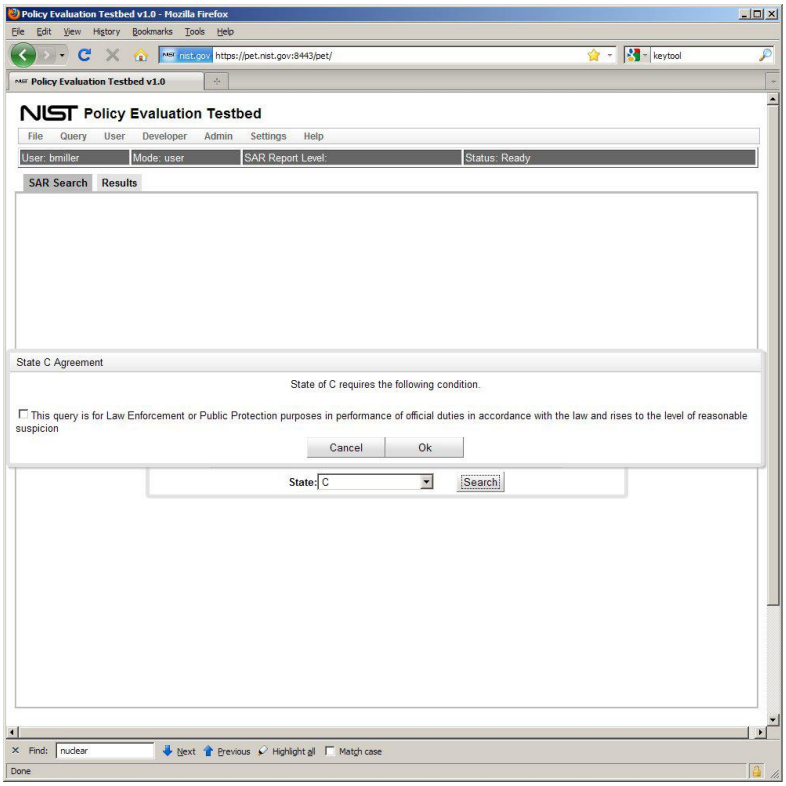

Figure 14. Complying with State's policies.

After Miller acknowledges that he is in accordance with State C's policies, the PET system searches for records related to "Springfield Nuclear Power Plant" in state C's repository. The SAR records retrieved are shown in Figure 15.

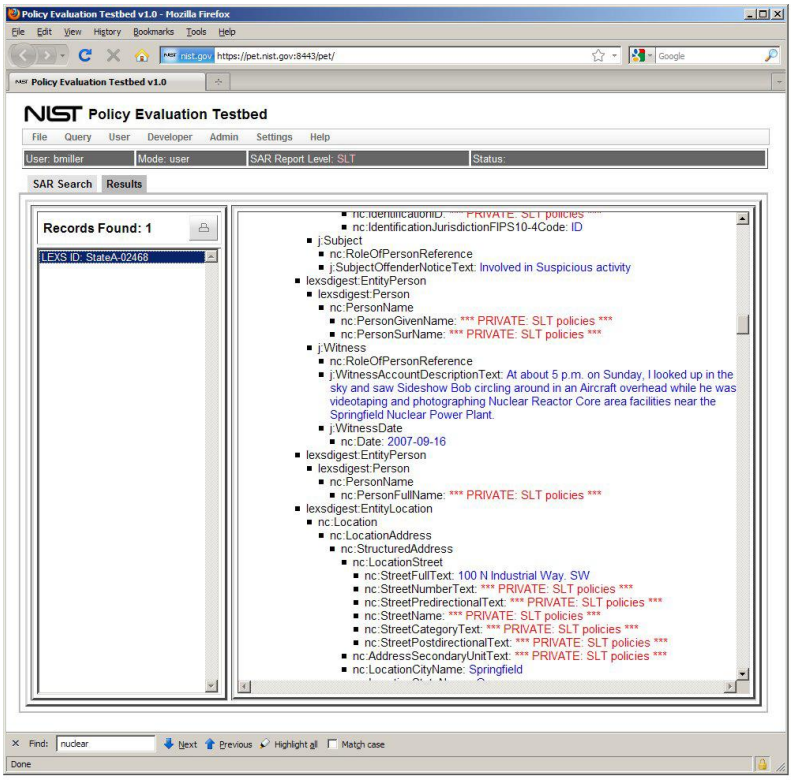

Figure 15. Retrieved summary SLT SAR record.

Here, we note that because Miller only has an assurance level of 2 and has remote access, he is not permitted to access full SAR records but only Summary SLT SAR records.

5.5.2 Full SAR Record Example

In the second scenario, Bob Richards (id: `brichards`, password: `brichards`) logs into the system and queries State B with a license number "`123456`" as shown in Figure 16. Richards then complies with State B's policies as shown in Figure 17.

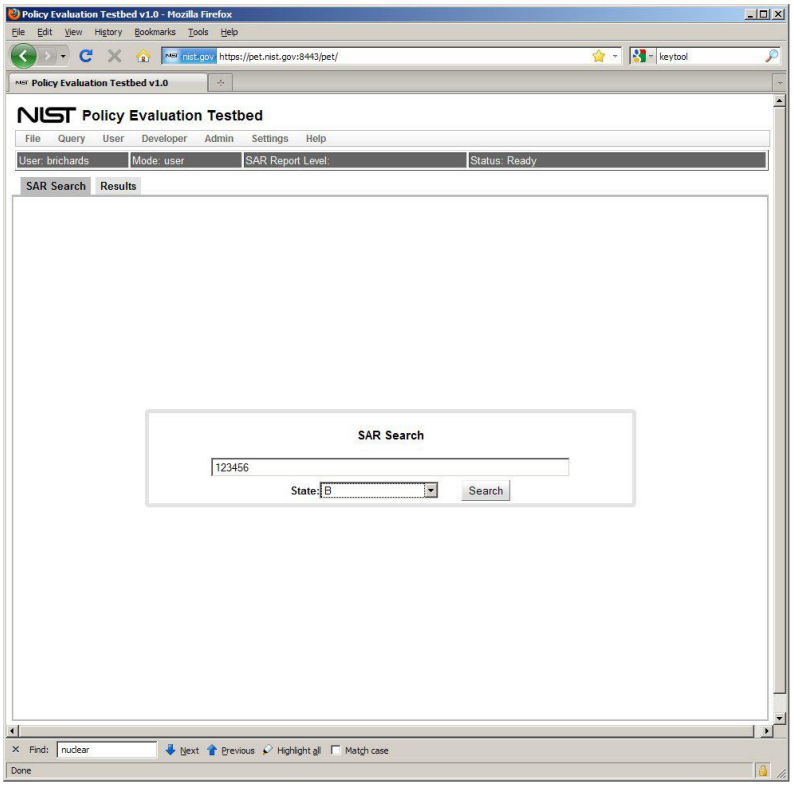

Figure 16. Querying license number.

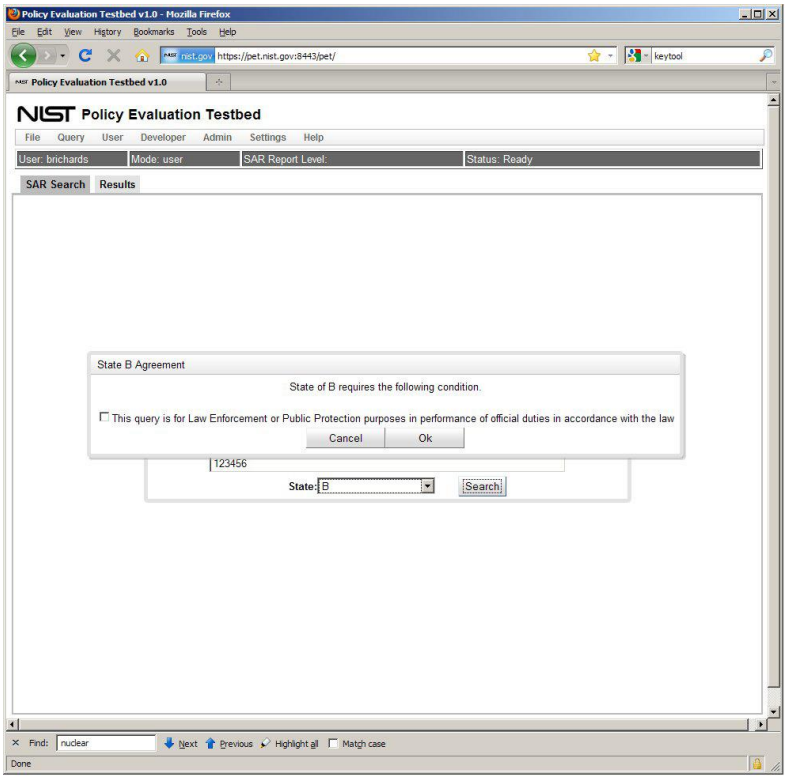

Figure 17. Complying with State B policies.

Richards is able to view Full SAR records from State B as shown in Figure 18.

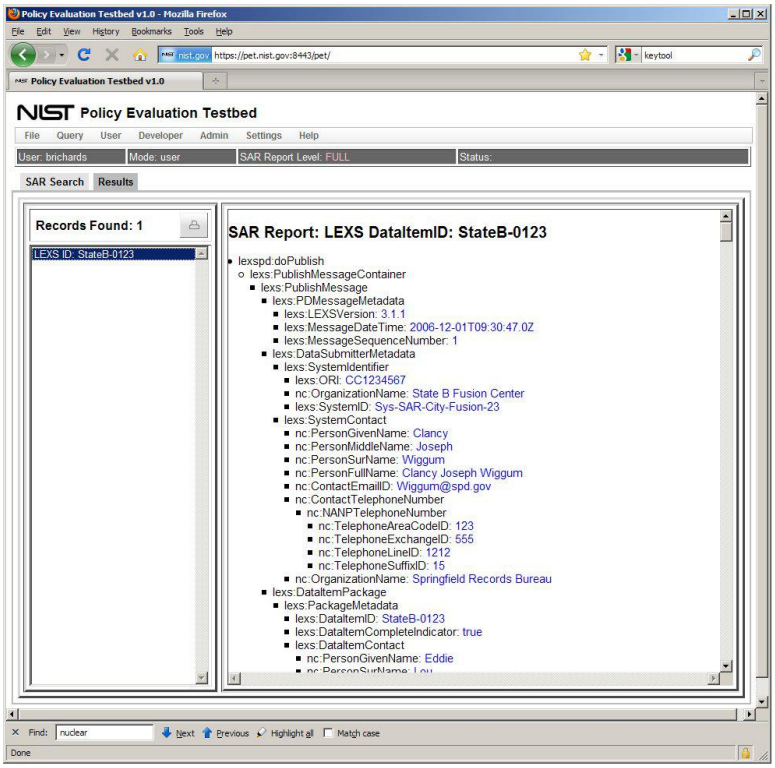

Figure 18. Retrieved full SAR report.

6 ISSUES AND RECOMMENDATIONS

In this section, we describe issues and recommendations related to each of the phases of the policy lifecycle as described in Section 2.

6.1 Policy Identification

6.1.1 Laws, Statutes and Policy Requirements

Issues The identification of laws, statutes and policies pertinent to granting or denying access to domain resources requires intimate knowledge of the domain resources, its users and the actions that users may invoke on resources as well as the pertinent federal, state, local, tribal, and domain-specific laws, statutes and policies. In general, access control policies are motivated by a need to ensure the privacy of information including classified data and personally identifiable information (PII).

Recommendations To facilitate the identification of laws, statutes and policies, this report recommends that a requirements analysis be conducted first to identify the general policy requirements of the domain resources including the conditions under which users are permitted or denied access to those resources. Such analysis should be conducted by experts that are intimately familiar with the users and resources of the domain as well as the actions that those users may invoke on those resources. After these general requirements are defined, work should be conducted to identify specific federal, state, local or domain-specific laws, statutes and policies that adequately address the identified requirements. For example, a general policy requirement in a law enforcement domain may be to ensure the privacy of PII data to all non-law enforcement personnel. After this general requirement is identified, experts may identify specific federal, state, local or domain-specific laws, statutes and policies that may apply. For example, 28 CFR Part 23 [DOJ], a federal regulation aimed at providing law enforcement with the guidance needed to operate criminal intelligence information systems effectively while safeguarding privacy and civil liberties, may be identified as an applicable policy.

6.1.2 Policy Derivation

6.1.2.1 Subject Attribute Data Model

Issues A subject attribute data model is necessary to derive a policy. Typically, a domain will have a general set of target users (e.g., law enforcement personnel) but may not have a defined subject attribute model for these users.

Recommendations To facilitate the derivation of a subject attribute model, this report recommends that during the identification phase of the policy lifecycle, an analysis be conducted to identify and analyze an existing subject attribute model that supports the requirements of the intended policies. For example, in a SAR domain, models including GFIPM or the Homeland Security Information Network (HSIN) [DHS] models should be

analyzed to determine if they define the subject attributes required to support access control policies for SAR data. In cases where an existing subject attribute model may not contain all of the attributes required, efforts should be made to extend the model through appropriate venues including workshops and standards groups. In cases where no such models exist, efforts should be made to collaborate with stakeholders to develop the specification for a new model. In this pilot, we analyzed the GFIPM specification within the context of the policies used to control access to SAR data. For these policies, we concluded that GFIPM provided the subject attributes required to support policies for controlling access to SAR data.

6.1.2.2 Resource Attribute Data Model

Issues A resource attribute data model is necessary to derive a policy. Typically, a domain will have a set of target resources (e.g., SAR records) but may not have a defined resource attribute model for those resources.

Recommendations To facilitate the derivation of a resource attribute data model, this report recommends that during the identification phase of the policy lifecycle, work should be conducted to identify and analyze an existing resource attribute model that supports the requirements of the intended policies. For example, in a SAR domain, models including the NIEM SAR [NIEM-b] should be analyzed to determine if they define the resource attributes required to support access control policies for SAR data. In cases where an existing resource attribute model may not contain all of the required attributes, efforts should be made to extend the model through appropriate venues including workshops and standards groups. In cases where no such models exist, efforts should be made to collaborate with stakeholders to develop the specification for a new model. In this pilot, we analyzed the NIEM SAR specification within the context of the identified policies used to control access to SAR data. We found that the level of meta-data for SAR record fields to be lacking. Such meta-data, including tags for indicating PII fields, could facilitate the mapping of laws, statutes and policies to control data-level access to SAR. Here, we recommend the addition of meta-data tags to indicate classifications of SAR fields to support rendering of policy decisions.

6.1.2.3 Environment Attribute Data Model

Issues An environment attribute data model may be necessary to derive a policy. An environment attribute data model specifies the environment attributes required by a policy to render a policy decision. Such attributes may include, for example, the date and time of a decision request or the IP address from which an access request is made.

Recommendations To facilitate the derivation of environment attribute data models, this report recommends that work should be conducted to identify and analyze an existing environment attribute model that supports the requirements of the intended policies. In cases where an existing environment attribute model may not contain all of the required attributes, efforts should be made to extend the model through appropriate venues including workshops and standards groups. In cases where no such models exist, efforts

should be made to collaborate with stakeholders to develop the specification for a new model.

6.1.2.4 Parsing Laws, Statutes and Policies

Issues During the identification of laws, statutes and policies from within their original documents, work must be conducted to parse natural language phrases, sentences and paragraphs into a form that may be mapped to ABAC rules. This process requires intimate knowledge of the laws, statutes and policies as well as the language used to express them.

Recommendations To facilitate the parsing of laws, statutes and policies, this report recommends that an expert fluent in the laws, statutes and policies examine the system's subject attributes and map the access privileges to the system's resources according to the rules of the applied laws, statutes and policies. This work is painstaking and costly because it requires experts to comprehend not only the laws, statutes and policies but also the semantics surrounding one or more heterogeneous system's subject and resource attributes. This report recommends a portable AC framework that is general enough to be used by AC administrators to compose their AC policies without the extra cost of translating or learning the laws/statutes. This General Attribute Framework (GAF) should be constructed from the content and ontology of the intended law/statute using Generic Attributes (GA), which can be applied to the specific attributes of any information system in different application domains. The goal of the GAF is to provide a framework as a layer between high-level laws /statutes and low-level policies/rules. GAF allows AC policy authors to compose AC policies without the expert knowledge of the law/statute. The knowledge of the law/statute is required only at the time when the framework is constructed. Derived from analyzing the content and ontology of the law/statute rules, the GAF contains access rules associated with the user and resource GAs, which are generic for any domain of an AC system. Therefore, a GAF is an AC policy with rules in terms of GA based on ABAC: subject/resource attributes, and actions. The format of a GAF AC rule is:

IF $\langle subjectGA_1 \rangle$ … AND $\langle subjectGA_N \rangle$ THEN ALLOW $\langle action_1 \rangle$ …AND $\langle action_n \rangle$ ACCESS TO RESOURCE WITH $\langle resourceGA_1 \rangle$ … AND $\langle resourceGA_N \rangle$

GAF will provide clear definitions and descriptions of the GAs by using common vocabulary such that any AC policy administrator who does not have the law/statute knowledge can understand. To enforce the law/statute on the information system, the AC policy administrator only need to assign the GAF's GAs as tags or meta-attributes to the users and resource by reviewing the existing user and resource attributes from the system. There is no need to create AC policy rules, which are already embedded in the GAF.

6.2 Policy Encoding

6.2.1 Policy Model Support

Issues The representation for encoding policies may affect the ability to support certain types of policies. For example, XACML does not support historical or state-based access control model [Alm] or descriptions of attribute relations [Hu-a].

Recommendations To ensure that an encoding representation supports the intended policies, an analysis should be conducted to determine the type of supported policy models. This entails the identification of all access control models required by the privilege management system.

6.2.2 Decision Request and Response

The representation for encoding policies may constrain the way in which the policies are queried by a privilege management system.

Recommendations To ensure that an encoding representation supports the desired query mechanism, this report recommends that an analysis be conducted to identify how the encoding restricts both decision queries and responses. For example, if policies are represented by a semantic language (e.g., OWL), then querying against this representation will require a specific query language (e.g., SPARQL [W3C-c]). It is also recommended to ensure that decision responses are inline with the system's requirements. For example, XACML-based systems typically only allow four responses to a query: `permit`, `deny`, `indeterminate`, and `not applicable`. Thus, an analysis must be conducted to determine if such query responses conform to the privilege management system requirements.

6.2.3 Policy Decision Point Requirements

Issues The representation for encoding policies is often closely coupled with a component or system infrastructure required for maintaining, querying, and processing encoded policies. This component is typically known as a Policy Decision Point (PDP). The selection of an encoding representation influences the type of PDP that will be required.

Recommendations To ensure the identification of an appropriate PDP for supporting the encoding representation, this report recommends that an analysis be conducted to assess commercial and open-source PDPs with respect to set-up, maintenance, performance, and cost. If the encoding representation is XACML, for example, a system component that is XACML-compliant, a query engine, and a messaging protocol will be necessary. An analysis must be conducted to determine if the PDP fits the architectural and implementation requirements of the intended privilege management system.

6.2.4 Policy Optimization

Issues Policies can be written in myriad ways and still allow for correct decision rendering. However, the way in which encoded policies are written affects processing efficiency and maintainability.

Recommendations To facilitate optimization of encoded policies, this report recommends that efforts should be made to encode the policies such that they are efficient and maintainable. For example, authors should write encoded policies using minimal lines of code while ensuring that the correctness and completeness of the policies remain intact.

6.3 Policy Testing

Issues After policies have been encoded, they must be tested to ensure that they are complete and correct. This is particularly true when multiple policies are considering when rendering a policy decision since discrepancies of conflicts may exist. Identifying discrepancies or conflicts between policy specifications and their intended function is crucial to the proper operation of a privilege management system. As a result, policy specifications must undergo rigorous verification and validation to ensure that the policy specifications truly encapsulate the desires of the policy authors.

Recommendations To ensure that policy specifications truly encapsulate the desires of policy authors, this report recommends that policies undergo rigorous verification and validation. Such testing can be accomplished by tools such as the NIST Access Control Policy Tool (ACPT) [Hu-b] tool. ACPT allows policy authors to conveniently specify access control models (such as RBAC and Multi-Level models) and rules as well as access control properties. From the specified models and rules, the ACPT tool automatically synthesizes deployable policies and comprehensively tests and verifies the policies against the specified properties, it then reports to the policy authors on the detected problems in the policies to prevent leaving security holes in the policies before deployment. ACPT tool provides policy authors high security confidence levels for their policy.

6.4 Policy Deployment

6.4.1 Policy Decision Point

Issues There are numerous issues related to the selection and use of a PDP. One issue concerns the level of compliance with the XACML specification and support for data retrieval. For example, some beta versions of PDP systems might not support obligations or might not have support to access certain database APIs in order to retrieve attributes (e.g., subject or resource attributes) needed to render a policy decision. In most cases, PDPs will be implemented as a "black box" that can be configured but cannot be easily

modified or extended to support additional (and often required) functionality. In the case of commercial PDP systems, a service contract might provide programming support for the PDP, but this likely will add substantial cost to a system that already requires very high upfront costs. On the other hand, open-source PDPs have little or no upfront costs, but modifying, extending, and maintaining the code may be nontrivial and will require a developer or administrator to maintain it.

Recommendations To assess the level of PDP-related development, as well as the cost for deploying and maintaining a PDP, this report recommends that a thorough requirements and budget analyses should be conducted. This report also recommends that an analysis be conducted to assess limitations with respect to features, functionality and compliance with XACML specification be conducted for both commercial and open-source PDP implementations.

6.4.2 Context Handler

Issues The context handler is the entity that connects the PDP to the rest of the privilege management system. Unlike the PDP, the context handler is typically designed and implemented from scratch to support the policy domain. There are a number of issues concerning the design and implementation of the context handler. One issue concerns the proper mapping and communication of decision requests and responses between the PDP and PEP. Another issue concerns the ability of the context handler to resolve queries by the PDP for user or resource attributes. Such queries are needed by the PDP to render policy decisions.

Recommendations To support the correct submission of decision requests by the PEP, this report recommends that the developer of the PEP have intimate familiarity with the PDP messaging and protocol API as well as the policies supported by the PDP. The context handler will require the development of code to support the communication protocols required by the PDP (e.g., SOAP and SAML) as well as the format of the XACML Request and Response formats. The design of the context handler may also require familiarity with the policies contained within the PDP in order to ensure that the context handler provides the PDP with sufficient information (i.e., attributes) to render a policy decision. This is particularly true if the PDP does not have sufficient capability to retrieve attributes required to resolve a decision request and the context handler will be required to provide these via a *Policy Information Point* (PIP), which is an entity that acts a source of attribute values.

6.4.3 Policy Enforcement Point

Issues Like the context handler, the Policy Enforcement Point (PEP) is a system entity that must be designed and implemented for the domain. There are several key issues that must be addressed when designing and implementing the PEP. One issue concerns the design of decision requests that are submitted to the PDP. Often the design of the queries will be dependent upon how the policies are encoded by the PDP. For example, if the PDP is XACML-based, then the PEP must use queries that result in one of four XACML

Responses: `permit`, `deny`, `indeterminate` and `not applicable`. Thus, the PEP must ensure that it expects only these types of responses. Another issue concerns the filtering of accessible data. In many cases, a policy will permit access to only parts of a resource (e.g., fields of a database record). In such cases, the PEP must be designed to either acquire only these permitted parts or filter the non-permitted parts from the resource before returning the results back to the requester. Another issue concerns the integration of the PEP with the interface for acquiring requests from an access requester.

Recommendations To ensure that queries submitted to the PDP (via the context handler) will result in a correct response, this report recommends that an analysis of the intended queries be conducted to identify all expected responses. An analysis should also be conducted with respect to data-level permissions defined by policies supported by the PDP to ensure that they can be properly enforced. Here, the PEP will require details about these permissions in order to enforce access control. For example, policies that are used to control access to medical records may allow some users to view an entire medical record while allowing others to view only specific fields. The PEP must be able to acquire information about which sets of fields are permissible and not permissible under certain policies. Such information can be obtained through a separate source (e.g., file or database) associated with the policies. Finally, the PEP must provide an API to allow access requesters a facility to submit requests for resources and retrieving permitted resources.

6.4.4 Access Requester

Issues In the PET system, the access requester is a web interface that allows users to request and display resources. The design and implementation of the access requester involves a number of issues. One issue concerns the interface design required to capture the resources required by the user. Another issue is the support for policies where attribute values must be acquired from the user during run-time. Such attribute values are likely not available via a PIP or from the resource. For example, some policies may require run-time acknowledgement from the user that the submitted query is within the context of his/her official duties. Support for this dynamic attribute acquisition requires the PEP and Access Requester developer to be familiar with familiar with the policies supported by the PDP. The design and implementation must also support the display of filtered results and PDP error responses such as `indeterminate` or `not applicable`.

Recommendations To facilitate development of the Access Requester component, this report recommends that an analysis be conducted to identify all of the types of resources that may be queried by users via the privilege management system and that the design and implementation of the access requester permit for the selection of these resources. If the availability of the resources changes, the design should allow for these changes to be rendered automatically in the requester interface. It is also recommended that an analysis of the policies be conducted to identify those policies that require attributes to be acquired from the user during run-time. Once these attributes are identified, a suitable user interface must be developed for capturing these attribute values. Finally, it is

recommended that an analysis be conducted to identify how resources must be rendered, particularly when rendering filtered data. For example, hospital records with non-permitted fields should either be blocked completely from view or a message specifying the policy that prevents their view should be displayed.

6.4.5 Policy Information Point

Issues The Policy Information Point (PIP) is the source for subject, environment, or resource attribute values and typically exists as a database or service. Like the PDP, a PIP is typically a commercial or open-source entity that requires integration with the privilege management system via an API or communication protocol. Issues surrounding the use of a PIP include ensuring that the PIP is populated with attribute values that conform to the intended subject, environment, and resource attribute models and that PIP is maintained by an authorized source. Other issues include the performance of the PIP and the ease of installing, configuring, and maintaining the PIP.

Recommendations To facilitate identification of an optimal Policy Decision Point, this report recommends that an analysis be conducted to assess the development, maintenance, and costs associated with both commercial and open-source PIP systems. In many cases, a privilege management system will be required to use a PIP maintained by some authoritative source (e.g., an agency's human resource office).

7 FUTURE RESEARCH AND DEVELOPMENT

This work has highlighted the need for several areas of research and development aimed at developing technologies for facilitating privilege management and data-level access control of SAR like systems.

7.1 Rule and Regulation Extraction

Creation of machine-readable policy is predicated on analyzing various sources (legal documents, organizational guidelines, etc.) to extract the rules and regulations. This work is done manually today, without tooling support. Results are captured, merged, and maintained independently of the sources. To remedy this situation, we envisage the development of Microsoft Word-based tooling to allow markup of documents and spreadsheets. The markup is then converted to a machine-readable format, based on the W3C Semantic web standards. This approach improves the capture experience, allows analysis of the machine-readable definitions for consistency and completeness, and maintains traceability of the resultant machine-readable rules to their sources. In addition, via import capabilities, existing vocabularies and rule definitions could be reused.

7.2 Vocabulary and Rule Sets

Information-sharing is predicated on the correct handling of information by both the sender and the receiver. In order to guarantee this, the vocabularies and rules of the sender and receiver must be understood and aligned. Today, this is accomplished via manual techniques such as database staging tables for data translation, and explicit code-based translation of rules. These approaches are difficult to understand and capture in code, and difficult to adapt as the parallel systems evolve. Also, the mappings are prone to errors based on lack of understanding of the domains, and due to the sheer amount of information that must be translated. Semantic web technologies are well-suited to the task of ontology mapping and alignment, such as GAF. We envisage the development of a mapping/merging environment for peer, subset and superset vocabularies and rule sets. The environment coordinates existing Semantic web tooling to present alignment suggestions to domain experts in order to iteratively build a valid map/merge scheme. Such work would analyze existing ontology alignment tools and explores the efficacy of natural language feedback.

8 REFERENCES

[Alm] Alm, C. , Wolf, R. and J. Posegga, "The OPL Access Control Policy Language", Trust, Privacy and Secuirty in Digital Business, Springer Berlin/Heidelberg Volume 5695, 2009.

[Blaze] M. Blaze, J. Feigenbaum, J. Ioannidis, "The KeyNote Trust-Management System Version 2", IETF RFC 2704, http://www1.cs.columbia.edu/~angelos/Papers/rfc2704.txt, September 1999.

[Brown] K. Brown, "Exploring Claims-Based Identity", http://msdn.microsoft.com/enus/magazine/cc163366.aspx.

[DHS] U.S. Department of Homeland Security, DHS Exhibit 300 Public Release BY10 / A&o – Homeland Security Information Network (HSIN), 2010.

[DOJ] 28 Code of Federal Regulations (CFR) Part 23, U.S. Department of Justice, 1980.

[GFIPM] Global Justice Information Sharing Initiative (Global) Security Working Group, Global Federated Identity and Privilege Management (GFIPM) Metadata Overview Version 1.0, 2008.

[GWT] Goole Inc., Google Web Toolkit, http://code.google.com/webtoolkit/.

[Hu-a] Hu, V., Ferraiolo, D. and S. I. Gavrila, "Specification of Attribute Relations for Access Control Policies and Constraints Using Policy Machine", the "Sixth International Conference on Information Assurance and Security" (IAS 2010). Atlanta, US, Aug, 23-25, 2010.

[Hu-b] Hu, V., Access Control Policy Test (ACPT) Tool, http://csrc.nist.gov/groups/SNS/acpt/index.html, 2010.

[Meier] Meier, W. eXist-db, http://exist.sourceforge.net/.

[NIEM-a] NIEM Program Manager Office, NIEM Introduction, 2007.

[NIEM-b] NIEM Program Manager Office, Nationwide Suspicious Activity Reporting (SAR) Initiative (NSI): NIEM Standards in Action, 2009.

[OASIS-a] OASIS, eXtensible Access Control Markup Lanaguage (XACML) Version 3.0, 2010.

[OASIS-b] OASIS, Security Assertion Markup Language (SAML) Version 2.0, 2005.

[Pimlott] A. Pimlott and O. Kiselyov, "Soutei, a Logic-Based Trust-Management System", FLOPS 2006, 8th International Symposium on Functional and Logic Programming. Fuji-Susono, Japan, April 24-26, 2006. Also in Springer's Lecture Notes in Computer Science 3945/2006, pp. 130-145.

[Saldhana] Saldhana, A. PicketBox XACML, https://community.jboss.org/wiki/PicketBoxXACMLJBossXACML, 2010.

[W3C-a] World Wide Web Consotium, Resource Description Framework (RDF), 2004.

[W3C-b] World Wide Web Consortium, OWL Web Ontology Language, 2004.

[W3C-c] World Wide Web Consortium, SPARQL Query Language for RDF, 2008.

[W3C-d] World Wide Web Consortium, Simple Object Access Protocol (SOAP), 2003.

[W3C-e] World Wide Web Consortium, XML Path Language (XPATH), 1999.

[W3C-f] World Wide Web Consortium, XQuery 1.0: An XML Query Language, 2007.