Attribute Based Access Control

Executive Summary

- Attribute based access control (ABAC) is an advanced method for managing access rights for people and systems connecting to networks and assets. Its dynamic capabilities offer greater efficiency, flexibility, scalability and security than traditional access control methods, without burdening administrators or users. In fact, Gartner recently predicted that “by 2020, 70% of enterprises will use attribute-based access control ... as the dominant mechanism to protect critical assets, up from less than 5% today.”¹

- Despite federal guidance that comprehensively defines ABAC and the considerations for enterprise deployment², adoption of ABAC has been slow.

- The National Cybersecurity Center of Excellence (NCCoE) addressed this challenge by developing an example ABAC reference model using commercial products that can be included alongside those in your existing infrastructure.

- The ABAC solution provided by this “How to” guide incorporates relevant security characteristics, standards, and best practices from the National Institute of Standards and Technology (NIST) and other organizations.

- The guide demonstrates the implementation of standards-based cybersecurity technologies in the real world. It can save organizations research and proof of concept costs for mitigating risk through the use of context for access decisions.

THE CHALLENGE

Traditionally, granting or revoking access to IT systems or other networked assets requires an administrator to manually enter information into a database-perhaps within several systems. This method is inefficient and doesn’t scale as organizations grow, merge, or reorganize. Further, this approach may not be best for preserving privacy and security: all users of a database have access to all its information, or administrators must limit access by constructing groups with specific permissions.

Consider a patient submitting a health insurance claim. A claims examiner needs to know just billing and diagnostic codes and a few pieces of demographic data in order to permit reimbursement. Interacting with the same system, the patient’s doctor needs to verify that the diagnosis and referral information is for the correct patient, but doesn’t need to see payment or address information. The patient needs access to the claim’s status, while the patient’s employer only needs to see the number of claims submitted by the employee. The insurance company provides a single service, claims processing, but each user of the service has different access needs.

An advanced method of access management would increase security and efficiency by seamlessly limiting some users’ views to more granular data. It would enable the appropriate permissions and limitations for the same information system for each user based on individual attributes, and allow for permissions to multiple systems to be managed by a single platform, without a heavy administrative burden.

²National Institute of Standards and Technology Special Publication (SP) 800-162, Guide to Attribute Based Access Control (ABAC) Definition and Considerations
THE SOLUTION

The NCCoE, part of NIST, demonstrated an advanced method, attribute based access control (ABAC), that uses granular attributes such as title, division, certifications and training—rather than a person’s role—to authorize an individual’s access. Access to an organization’s network or assets can be made based on information that is available to systems across an organization, or among organizations, about a person, the action she wants to execute, and the resource she wants to access. An orthopedist responding to a mass casualty event in a neighboring state can quickly gain access to a hospital’s patient records and radiology and pharmacy ordering systems, and only to those systems, based on authentication of her credentials and attributes such as employee status, medical specialization, and certifications. Additional visiting orthopedists are immediately granted the same permissions based on the same rules.

ABAC offers efficiencies and enhanced security in non-emergency scenarios, too. ABAC can provide separation of duties to help guard against fraud: a car insurance claims adjuster, for example, can be permitted to enter data about damage and generate a check, but only his supervisor can electronically sign the check. In addition to authorizing people, ABAC can be used to efficiently manage access among networked tools, devices, and systems that request access to corporate resources like applications, networks, systems, and data.

The NIST Cybersecurity Practice Guide Attribute Based Access Control shows how commercially available technologies can meet your organization’s needs to make access decisions for a diverse set of people and things, including those seeking access from external organizations. The complete guide is available at http://nccoe.nist.gov.

Approach

In our lab at the NCCoE, we simulated a typical electronic file library with a diverse set of resources from different divisions in an organization. Different files have different security levels.

We demonstrated how detailed attributes can be assigned to users and networked resources, and how fine-grained environmental considerations like time of day or IP address can provide context for access decisions, allowing for more informed, finely-tuned access decisions that increase security.

The guide:
- maps security characteristics to guidance and best practices from NIST and other standards organizations
- provides
  - a detailed example solution with capabilities that address security controls
  - instructions for implementers and security engineers, including examples of all the necessary components and their installation, configuration, and integration
- uses products that are readily available and interoperable with existing information technology (IT) infrastructure and investments
- is suitable for organizations of all sizes

While we have used a suite of commercial products to address this challenge, this guide does not endorse these particular products, nor does it guarantee regulatory compliance. Your organization’s security experts should identify the standards-based products that will best integrate with your existing tools and IT system infrastructure. Your organization can adopt this solution, or one that aligns to these guidelines, in whole, or you can use this guide as a starting point for tailoring and implementing parts of a solution.
BENEFITS

Our example solution:

- allows products and capabilities to be adopted on a component-by-component basis, or as a whole
- supports organizations with a diverse set of users and access needs, offering efficiencies in provisioning access
- reduces the number of identities managed by the enterprise, thereby reducing costs
- enables a wider range of risk-mitigation decisions by allowing organizations to define attribute-based policies for users and networked devices that include factors such as environment and time of day
- supports collaboration among organizations by allowing an enterprise to accept identities authorized by other enterprises, eliminating the need to pre-provision access for those identities
- supports the centralization of auditing and access policy management, creating efficiencies of policy management and reducing the complexity of regulatory compliance

SHARE YOUR FEEDBACK

You can get the guide at http://nccoe.nist.gov and help improve it by contributing feedback. As you review and adopt this solution for your own organization, we ask you and your colleagues to share your experience and advice with us. We recognize that technical solutions alone will not fully enable the benefits of ABAC, so we encourage organizations to share lessons learned and best practices for transforming the business processes associated with implementing ABAC.

- email abac-nccoe@nist.gov
- participate in our forums at https://nccoe.nist.gov/forums/attribute-based-access-control

Or learn more by arranging a demonstration of this reference solution by contacting us at abac-nccoe@nist.gov

TECHNOLOGY PARTNERS

The NCCoE designed and implemented this project with its National Cybersecurity Excellence Partnership (NCEP) partners.

Microsoft

Nextlabs®

PingIdentity®

RSA

Symantec®

The National Cybersecurity Center of Excellence at the National Institute of Standards and Technology addresses businesses’ most pressing cybersecurity problems with practical, standards-based example solutions using commercially available technologies. As the U.S. national lab for cybersecurity, the NCCoE seeks problems that are applicable to whole sectors, or across sectors. The center’s work results in publicly available NIST Cybersecurity Practice Guides that provide modular, open, end-to-end reference designs.
ATTRIBUTE BASED ACCESS CONTROL

For CIOs, CISOs, and Security Managers

Approach, Architecture, and Security Characteristics

Bill Fisher     Norm Brickman     Santos Jha
Sarah Weeks    Ted Kolovos      Prescott Burden

Leah Kauffman, Editor-in-Chief

NIST SPECIAL PUBLICATION 1800-3b
ATTRIBUTE BASED ACCESS CONTROL
DISCLAIMER

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST or NCCoE, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback. All publications from NIST’s National Cybersecurity Center of Excellence are available at http://nccoe.nist.gov.

Comments on this publication may be submitted to: abac-nccoe@nist.gov

Public comment period: September 30, 2016 through December 4, 2016

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
100 Bureau Drive (Mailstop 2002) Gaithersburg, MD 20899
Email: abac-nccoe@nist.gov
NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE) at the National Institute of Standards and Technology (NIST) addresses businesses’ most pressing cybersecurity problems with practical, standards-based solutions using commercially available technologies. The NCCoE collaborates with industry, academic, and government experts to build modular, open, end-to-end reference designs that are broadly applicable and repeatable. The center’s work results in publicly available NIST Cybersecurity Practice Guides, Special Publication Series 1800, that provide users with the materials lists, configuration files, and other information they need to adopt a similar approach.

To learn more about the NCCoE, visit http://nccoe.nist.gov. To learn more about NIST, visit http://www.nist.gov.

NIST CYBERSECURITY PRACTICE GUIDES

NIST Cybersecurity Practice Guides (Special Publication Series 1800) target specific cybersecurity challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the adoption of standards-based approaches to cybersecurity. They show members of the information security community how to implement example solutions that help them align more easily with relevant standards and best practices.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. The documents in this series do not describe regulations or mandatory practices, nor do they carry statutory authority.

ABSTRACT

Enterprises rely upon strong access control mechanisms to ensure that corporate resources (e.g. applications, networks, systems and data) are not exposed to anyone other than an authorized user. As business requirements change, enterprises need highly flexible access control mechanisms that can adapt. The application of attribute based policy definitions enables enterprises to accommodate a diverse set of business cases. This NCCoE practice guide details a collaborative effort between the NCCoE and technology providers to demonstrate a standards-based approach to attribute based access control (ABAC).

This guide discusses potential security risks facing organizations, benefits that may result from the implementation of an ABAC system and the approach that the NCCoE took in developing a reference architecture and build. Included is a discussion of major architecture design considerations, explanation of security characteristic achieved by the reference design and a mapping of security characteristics to applicable standards and security control families.

For parties interested in adopting all or part of the NCCoE reference architecture, this guide includes a detailed description of the installation, configuration and integration of all components.
KEYWORDS
access control; access management; attribute provider; authentication; authorization; identity federation; identity management; identity provider; relying party

ACKNOWLEDGMENTS
We gratefully acknowledge the contributions of the following individuals and organizations for their generous contributions of expertise, time, and products.

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nate Lesser</td>
<td>NIST National Cybersecurity Center of Excellence</td>
</tr>
<tr>
<td>Paul Timmel</td>
<td>NIST National Cybersecurity Center of Excellence</td>
</tr>
<tr>
<td>Paul Grassi</td>
<td>NIST National Strategy for Trusted Identities in Cyberspace</td>
</tr>
<tr>
<td>Mike Garcia</td>
<td>NIST National Strategy for Trusted Identities in Cyberspace</td>
</tr>
<tr>
<td>Naomi Lefkovitz</td>
<td>NIST National Strategy for Trusted Identities in Cyberspace</td>
</tr>
<tr>
<td>Rene Peralta</td>
<td>NIST National Strategy for Trusted Identities in Cyberspace</td>
</tr>
<tr>
<td>Dave Ferriolao</td>
<td>NIST Computer Security Division</td>
</tr>
<tr>
<td>Vincent Hu</td>
<td>NIST Computer Security Division</td>
</tr>
<tr>
<td>Roger Wiggenstam</td>
<td>NextLabs Inc</td>
</tr>
<tr>
<td>John Conduit</td>
<td>NextLabs Inc</td>
</tr>
<tr>
<td>Srikanth Karanam</td>
<td>NextLabs Inc</td>
</tr>
<tr>
<td>Adam Madlin</td>
<td>Symantec Corporation</td>
</tr>
<tr>
<td>Steve Kruse</td>
<td>Symantec Corporation</td>
</tr>
<tr>
<td>Steve Schmalz</td>
<td>RSA</td>
</tr>
<tr>
<td>Ben Smith</td>
<td>RSA</td>
</tr>
<tr>
<td>Andrew Whelchel</td>
<td>RSA</td>
</tr>
<tr>
<td>Chris Leggett</td>
<td>Ping Identity</td>
</tr>
<tr>
<td>Paul Fox</td>
<td>Microsoft Corporation</td>
</tr>
<tr>
<td>Derek Keatley</td>
<td>Microsoft Corporation</td>
</tr>
<tr>
<td>Hemma Prafullchandra</td>
<td>Hytrust</td>
</tr>
<tr>
<td>John McLeese</td>
<td>Hytrust</td>
</tr>
<tr>
<td>Name</td>
<td>Organization</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Dave Cox</td>
<td>ID/Dataweb</td>
</tr>
<tr>
<td>Chris Donovan</td>
<td>ID/Dataweb</td>
</tr>
</tbody>
</table>
## Contents

### List of Figures ix

### List of Tables xi

#### 1 Summary ............................................................................................................................... 1

1.1 The Challenge..................................................................................................................... 2

1.2 The Solution......................................................................................................................... 3

1.3 Risks ..................................................................................................................................... 3

1.4 Benefits ................................................................................................................................. 3

1.5 Technology Partners............................................................................................................. 4

1.6 Feedback ............................................................................................................................... 4

#### 2 How to Use This Guide................................................................................................... 5

#### 3 Introduction..................................................................................................................... 9

3.1 Background......................................................................................................................... 10

3.2 ABAC and RBAC Considerations....................................................................................... 10

3.3 ABAC Leveraging Identity Federation.................................................................................. 11

3.4 Security Standards.............................................................................................................. 12

#### 4 Approach ......................................................................................................................... 15

4.1 Audience ........................................................................................................................... 16

4.2 Scope.................................................................................................................................. 16

4.2.1 Assumptions ................................................................................................................... 16

4.2.1.1 Modularity................................................................................................................... 16

4.2.1.2 Business Policy Language ......................................................................................... 17

4.2.1.3 Attribute Semantics and Syntax ................................................................................ 17

4.2.1.4 Attribute Provenance ................................................................................................. 17

4.2.1.5 Trust Relationships for Identity Federation ............................................................... 17

4.2.1.6 Human Resources Database/Identity Proofing .......................................................... 17

4.2.1.7 Technical Implementation .......................................................................................... 18

4.2.1.8 Limited Scalability Testing ......................................................................................... 18

4.3 Risk Assessment ................................................................................................................... 18

4.4 Security Characteristics and Controls Mapping ................................................................. 19

4.5 Technologies ........................................................................................................................ 21

#### 5 Architecture..................................................................................................................... 25

5.1 Overview ............................................................................................................................. 26

5.1.1 User Authentication and the Creation of an Authentication Context............................ 26

5.1.2 Federation of a User Identity and Attributes .................................................................. 26
List of Figures

Figure 5.1  ABAC Build 1 Architecture .............................................................................. 30
Figure 5.2  UML Sequence Diagram .................................................................................. 33
Figure 5.3  Secondary Attribute Request Flow ................................................................. 35
List of Tables

1. Table 3.1 Related Security Standards and Best Practices ......................................................... 12
2. Table 4.1 Use Case Security Characteristics Mapped to Relevant Standards and Controls .......... 19
3. Table 4.2 Use Case Security Characteristics Mapped to Relevant Build Products ........ 21
Chapter 1. Summary

1 Summary

1.1 The Challenge ............................................................................................................... 2
1.2 The Solution ................................................................................................................... 3
1.3 Risks ............................................................................................................................... 3
1.4 Benefits ......................................................................................................................... 3
1.5 Technology Partners .................................................................................................... 4
1.6 Feedback ....................................................................................................................... 4
Traditionally, granting or revoking access to IT systems or other networked assets requires an administrator to manually enter information into a database—perhaps within several systems. This method is inefficient and doesn’t scale as organizations grow, merge, or reorganize. Further, this approach may not be best for preserving privacy and security: all users of a database have access to all its information, or administrators must limit access by constructing groups with specific permissions.

Attribute based access control (ABAC) is an advanced method for managing access rights for people and systems connecting to networks and assets. Its dynamic capabilities offer greater efficiency, flexibility, scalability and security than traditional access control methods, without burdening administrators or users.

Despite ABAC’s advantages and federal guidance that comprehensively defines ABAC and the considerations for enterprise deployment, adoption has been slow. In response, the National Cybersecurity Center of Excellence (NCCoE), part of the National Institute of Standards and Technology (NIST), developed an example of an advanced access control system. Our attribute based access control (ABAC) solution can more securely and efficiently manage access to networked resources, and with greater granularity that traditional access management. It enables the appropriate permissions and limitations for the same information system for each user based on individual attributes, and allows for permissions to multiple systems to be managed by a single platform, without a heavy administrative burden.

Our approach uses commercially available products that can be included alongside your current products in your existing infrastructure.

This example solution is packaged as a “How To” guide that demonstrates implementation of standards-based cybersecurity technologies in the real world. It can save organizations research and proof of concept costs for mitigating risk through the use of context for access decisions.

### 1.1 The Challenge

Enterprises face the continual challenge of providing access control mechanisms for subjects requesting access to corporate resources (e.g. applications, networks, systems, and data). The growth and distributed nature of enterprise resources, increasing diversity in users, credentials, and access needs, as well as the need to share information among stakeholders that are not managed directly by the enterprise, has given rise to the demand for access control system that enables fine-grained access decisions based on a range of users, resources, and environmental conditions.

Consider a patient submitting a health insurance claim. A claims examiner needs to know just billing and diagnostic codes and a few pieces of demographic data in order to permit reimbursement. Interacting with the same system, the patient’s doctor needs to verify that the diagnosis and referral information is for the correct patient, but doesn’t need to see payment or address information. The patient needs access to the claim’s status, while the patient’s employer only needs to see the number of claims submitted by the employee. The insurance company provides a single service, claims processing, but each user of the service has different access needs.

An advanced method of access management would increase security and efficiency by seamlessly limiting some users’ views to more granular data. It would enable the appropriate permissions and limitations for
Chapter 1. Summary

The same information system for each user based on individual attributes, and allow for permissions to multiple systems to be managed by a single platform, without a heavy administrative burden.

1.2 The Solution

This document details our approach in developing a standards-based ABAC solution. Through discussions with identity and access management (IdAM) experts and collaborating technology partners, the NCCoE developed a set of security characteristics required to meet the IdAM risks facing today's enterprises. The NCCoE mapped security characteristics to standards and best practices from NIST and other standards organizations, then used products from our technology partners as modules in an end-to-end example solution that mitigates IdAM risks.

1.3 Risks

Access control systems implement a process for defining security policy and regulating access to resources such that only authorized entities are granted access according to that policy. They are fundamental to mitigating the risk of unauthorized access not only from malicious external users and insider threats, but also from acts of misfeasance. In the absence of a robust access control system, enterprises struggle to control and audit access to their most sensitive data and risk the loss or exposure of critical assets, loss of trust in employees and from customers, and harm to brand reputation.

As technology pervades all business processes, access control systems must support increasing diversity in users, credentials and access needs including digital identities from external security domains. This increases the overhead associated with managing access control systems and introduces increased risk of unauthorized access as organizational policies escalate in complexity.

At the strategic level, organizations face risks associated with the acquisition, deployment, and maintenance of access control systems. These risks include the cost of the implementation and maintenance, any compliance or regulatory requirements, as well as a lack of preceding implementations from which to derive lessons learned.

1.4 Benefits

The example solution described in this guide has the following benefits:

- products and capabilities can be adopted on a component-by-component basis, or as a whole
- supports organizations with a diverse set of users and access needs, reducing the risks of “privilege creep” (a user obtains access levels beyond those needed), and creating efficiencies in the provisioning of accesses
- reduces the number of identities managed by the enterprise, and thereby reduces costs associated with those management activities
- enable a wider range of risk-mitigation decisions by allowing organizations to define attribute-based policy on subjects and objects, but also using a variety of environmental decisions
- supports business collaboration, by allowing the enterprise to accept federated identities and eliminating the need to pre-provision access for identities being federated.
supports the centralization of auditing and access policy management, creating efficiencies of policy management and reducing the complexity of regulatory compliance.

1.5 Technology Partners

The NCCoE designed and implemented this project with its National Cybersecurity Excellence Partner (NCEP). NCEPs are IT and cybersecurity firms that have pledged to support the NCCoE’s mission of accelerating the adoption of standards-based, secure technologies. They contribute hardware, software, and expertise. In this project, we worked with:

- Ping Identity
- NextLabs
- Microsoft
- RSA
- Symantec

1.6 Feedback

A NIST Cybersecurity Practice Guide does not describe "the" solution, but a possible solution. This is a draft guide. As you review and adopt this solution for your own organization, we ask you and your colleagues to share your experience and advice with us. Your comments, suggestions, and success stories will improve subsequent versions of this guide.

- email abac-nccoe@nist.gov
- participate in our forums at https://nccoe.nist.gov/forums/attribute-based-access-control
- Or learn more by arranging a demonstration of this example solution by contacting us at abac-nccoe@nist.gov
2 How to Use This Guide
This NIST Cybersecurity Practice Guide demonstrates a standards-based example solution and provides users with the information they need to replicate this approach to identity and access management. The example solution is modular and can be deployed in whole or in part.

This guide contains three volumes:

- **NIST SP 1800-3a: Executive Summary**
- **NIST SP 1800-3b: Approach, Architecture, and Security Characteristics** – what we built and why (this document)
- **NIST SP 1800-3c: How-To Guides** – instructions for building the example solution

Depending on your role in your organization, you might use this guide in different ways:

- Business decision makers, including chief security and technology officers will be interested in the **Executive Summary (NIST SP 1800-3a)**, which describes the:
  - challenges enterprises face in implementing and using access control mechanisms
  - example solution built at the NCCoE
  - benefits of adopting ABAC, and the limitations of role based access (RBAC) systems

- Technology or security program managers who are concerned with how to identify, understand, assess, and mitigate risk will be interested in this part of the guide, **NIST SP 1800-3b**, which describes what we did and why. The following sections will be of particular interest:
  - Section 4.3, Risk Assessment, provides a detailed description of the risk analysis we performed.
  - Section 4.4, Security Characteristics and Controls Mapping, maps the security characteristics of this example solution to cybersecurity standards and best practices.

- You might share the **Executive Summary, NIST SP 1800-3a**, with your leadership team members to help them understand the importance of adopting standards-based access management approaches to protect your organization's digital assets.

- IT professionals who want to implement an approach like this will find the whole practice guide useful. You can use the How-To portion of the guide, **NIST SP 1800-3c**, to replicate all or parts of the build created in our lab. The How-To guide provides specific product installation, configuration, and integration instructions for implementing the example solution. We do not re-create the product manufacturers’ documentation, which is generally widely available. Rather, we show how we incorporated the products together in our environment to create an example solution.

- This guide assumes that IT professionals have experience implementing security products within the enterprise. While we have used a suite of commercial products to address this challenge, this guide does not endorse these particular products. Your organization can adopt this solution or one that adheres to these guidelines in whole, or you can use this guide as a starting point for tailoring and implementing parts of a solution that would support the deployment of an ABAC system and the corresponding business processes. Your organization’s security experts should identify the products that will best integrate with your existing tools and IT system infrastructure. We hope you will seek products that are congruent with

---
1. Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept. Such identification is not intended to imply recommendation or endorsement by NIST or the NCCoE, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.
applicable standards and best practices. Section 4.5, Technologies, lists the products we used and maps
them to the cybersecurity controls provided by this reference solution.

A NIST Cybersecurity Practice Guide does not describe “the” solution, but a possible solution. This is a
draft guide. We seek feedback on its contents and welcome your input. Comments, suggestions, and
success stories will improve subsequent versions of this guide. Please contribute your thoughts to abac-
nccoe@nist.gov, and join the discussion at https://nccoe.nist.gov/forums/attribute-based-access-control.
Chapter 3. Introduction

3 Introduction

3.1 Background .................................................................................................................. 10
3.2 ABAC and RBAC Considerations .................................................................................... 10
3.3 ABAC Leveraging Identity Federation .............................................................................. 11
3.4 Security Standards .......................................................................................................... 12
3.1 Background

Basic read, write, and execute permissions, along with discretionary access control (DAC) and mandatory access control (MAC) principles, mark the evolution of access control to the RBAC models that are in common commercial use today. While RBAC focuses primarily on the use of the role attribute, ABAC allows for access decisions based upon arbitrary attributes.

NIST SP 800-162, Guide to Attribute Based Access Control (ABAC) Definition and Considerations, describes ABAC as “a logical access control model that is distinguishable because it controls access to objects by evaluating rules against the attributes of” (a) the subject or user requesting access, (b) the target object for which access or a transaction is being requested, and (c) the environment relevant to a request. It continues:

“...In its most basic form, ABAC relies upon the evaluation of attributes of the subject, attributes of the object, environment conditions, and a formal relationship or access control rule defining the allowable operations for subject-object attribute and environment condition combinations. All ABAC solutions contain these basic core capabilities that evaluate attributes and environment conditions, and enforce rules or relationships between those attributes and environment conditions.”

“The rules or policies that can be implemented in an ABAC model are limited only to the degree imposed by the computational language. This flexibility enables the greatest breadth of subjects to access the greatest breadth of objects without specifying individual relationships between each subject and each object.”

In order to enable ABAC implementations, the standards community has undertaken efforts to develop common terminology and interoperability across access control systems. One such standard is the eXtensible Access Control Markup Language (XACML). Built on an eXtensible Markup Language (XML) foundation, XACML is designed to allow externalized, run-time access control decisions using attribute-based policy definitions.

3.2 ABAC and RBAC Considerations

RBAC simplifies identity management by grouping users with similar access needs by role. Privileges can then be assigned to a role rather than an individual user. This simplification has led to the almost ubiquitous adoption of the RBAC model for logical access control. However, in the modern IT environment, enterprises face growing diversity in both types of users and their access needs. This diversity elucidates several limitations of the RBAC model.

This diversity introduces a number of administrative and policy enforcement challenges. Administrators manage access policy for multiple applications and security domains, with each often requiring discrete access control policies. Most systems implement access control in different ways, making it hard to share...

information across systems and requiring administrators to configure the access for like users uniquely in each system, typically by using the roles or groups native to that system.

These roles are often insufficient in the expression of real-world access control policies and cannot handle real-time environmental considerations that may be relevant to access control decisions; examples such as the location of access, time of day, threat level, and client patch level illustrate how enterprises could be afforded a wider range of decisions based on the amount of risk they perceive or are willing to accept. Similarly, RBAC does not readily support attributes relating to authentication context, referring to assurance of a user’s login process.

Attribute-based systems, by the nature of their name:value pairs for each attribute, can support a much finer-grained authorization environment than an RBAC system. ABAC allows business logic to be translated into attribute-based policies that govern access decisions, allowing for a common and centralized way of expressing policy and computing and enforcing decisions, over the access requests for diverse systems. These policies include the ability to take environmental considerations into account when making access decisions.

Attribute policy definitions establish a relationship between subject and object that does not change as attribute values change, thus reducing the opportunity for privilege creep and maintaining separation of duties. ABAC systems have the ability to permit new types of access requests without the need to alter the current set of subject/object relationships. Instead, the enterprise can define a new attribute or attributes (or a combination of currently used attributes) that represents the new level of access needed and then define an attribute-based policy that supports this level of access.

3.3 ABAC Leveraging Identity Federation

As enterprises look to keep up with leading-edge technology solutions, they face the identity management challenge of allowing a diverse set of digital identities access to many different organizational applications and resources. Commonly, this requires recognizing digital identities from external security domains, which are typically trusted strategic business stakeholders. Enterprises have realized that supporting this wide range of users, which may not be known or managed by the enterprise, requires attributes from external sources. One approach to meeting this requirement uses federation profiles.

Identity federation profiles define the methods used to convey a set of user information from the Identity Provider (IdP), or organization where the user is known, to the target location or Relying Party (RP) that needs to acquire the information for some use such as access control. These technologies leverage widely accepted, open, Web-oriented standardized communication languages, like the Security Assertion Markup Language (SAML) version 2.0 standard from OASIS¹, which uses XML, or the OpenID Connect (OIDC) standard from the OpenID Foundation² built upon JavaScript Object Notation (JSON), to carry the assertions about a user. Federation profiles allow identity and attribute information to be sent over Hypertext Transfer Protocol (HTTP) in a manner that can be understood and used by the receiving organization (the RP) to make access control decisions.

In some cases an RP may need to obtain attributes about a user from a source other than the user’s IdP. In this case the RP may receive a user’s attributes from a trustworthy external source known as an Attribute Provider (AP). Commonly, identity federation profiles are used to facilitate the federation of attributes from the AP to the RP.

Enterprises looking to participate in federation must have a degree of trust in the organization from which they are receiving identity and attribute information. To facilitate these trust relationships, non-profit organizations such as the Kantara Initiative and the Open Identity Exchange (OIX) have proposed or issued trust framework specifications that provide a set of contracts, regulations, and commitments. These specifications enable parties to a trust relationship to rely on identity and attribute assertions (via federation profiles) from external entities.

Identity federation allows external users to gain access to Web-based protected resources, without the need for the RP to manage the identity. When identities and access decisions are abstracted into a common set of attributes, access decisions can be externalized and policies can be established across business units or even organizational boundaries. Identity and attribute federation enables access decisions for users from trusted IdPs, even if the users have not previously been provisioned by the RP (sometimes referred to as the “unanticipated user” scenario).

### 3.4 Security Standards

<table>
<thead>
<tr>
<th>Related Technology</th>
<th>Relevant Standard</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS Institute, Critical Security Controls</td>
<td></td>
<td><a href="https://www.sans.org/critical-security-controls/">https://www.sans.org/critical-security-controls/</a></td>
</tr>
<tr>
<td>ISACA, COBIT 5</td>
<td></td>
<td><a href="http://www.isaca.org/COBIT/Pages/Product-Family.aspx">http://www.isaca.org/COBIT/Pages/Product-Family.aspx</a></td>
</tr>
<tr>
<td>Cloud Security Alliance, Cloud Controls Matrix v3.0.1</td>
<td></td>
<td><a href="https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3-0-1/">https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3-0-1/</a></td>
</tr>
<tr>
<td>Related Technology</td>
<td>Relevant Standard</td>
<td>URL</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Access Control (ABAC)</td>
<td>NIST SP 800-162, Guide to Attribute Based Access Control (ABAC) Definition and Considerations</td>
<td><a href="http://dx.doi.org/10.6028/NIST.SP.800-162">http://dx.doi.org/10.6028/NIST.SP.800-162</a></td>
</tr>
<tr>
<td>Language (OIDC)</td>
<td>OpenID Connect Core 1.0</td>
<td><a href="http://openid.net/specs/openid-connect-core-1_0.html">http://openid.net/specs/openid-connect-core-1_0.html</a></td>
</tr>
<tr>
<td>Language (SAML)</td>
<td>OASIS Security Assertion Markup Language (SAML) V2.0</td>
<td><a href="http://saml.xml.org/samlSpecifications">http://saml.xml.org/samlSpecifications</a></td>
</tr>
<tr>
<td>Language (WS-Federation)</td>
<td>OASIS Web Services Federation Language (WS-Federation) Version 1.2</td>
<td><a href="http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html">http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html</a></td>
</tr>
</tbody>
</table>
Table 3.1  Related Security Standards and Best Practices (Continued)

<table>
<thead>
<tr>
<th>Related Technology</th>
<th>Relevant Standard</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI</td>
<td>PKI Technical Standards</td>
<td><a href="http://www.oasis-pki.org/resources/techstandards/">http://www.oasis-pki.org/resources/techstandards/</a></td>
</tr>
</tbody>
</table>
Chapter 4. Approach

4 Approach

4.1 Audience

4.2 Scope

4.3 Risk Assessment

4.4 Security Characteristics and Controls Mapping

4.5 Technologies
4.1 Audience

This guide is intended for individuals responsible for implementing IT security solutions.

4.2 Scope

This project began with discussions between the NCCoE, identity and access management experts across NIST, and IT security vendors partnered with the NCCoE. These discussions enumerated an array of technologies and standards relevant to the ABAC space, but very few implementations of ABAC technology.

In response, the NCCoE drafted a white paper\(^1\) that identified numerous desired solution characteristics. After two rounds of public comments on the document, the NCCoE worked with its NCEP to design an architecture that would demonstrate an array of ABAC capabilities. This build does not include every characteristic found in the white paper, but does include the relevant set of ABAC capabilities\(^2\) based on the technology available to us through the portfolios of the NCCoE’s National Cybersecurity Excellence Partners. The scope of this build is the successful execution of the following capabilities:

- identity and attribute federation between trust partners
- user authentication and creation of an authentication context
- fine-grained access control through a policy enforcement point (PEP) closely coupled with the application
- creation of attribute-based policy definitions
- secondary attribute requests
- allowing RP access decisions on external identities without the need for pre-provisioning

4.2.1 Assumptions

The ABAC build described here incorporates the assumptions in this section.

4.2.1.1 Modularity

This example solution is made of many commercially available parts. You might swap one of the products we used for one that is better suited for your environment. We also assume that you already have some IdAM solutions in place. The use of standard protocols such as SAML, LDAP, and WS-Federation enhances the modularity of the architecture to improve your identity and access/authorization functions without major impact to your existing infrastructure. For organizations that want to limit their ABAC deployment


\(^2\)This project has the overarching goal of demonstrating technical implementations of standards-based ABAC functionality. In enumerating technology relevant to this effort, we worked closely with experts from the identity and access management community. During those discussions, we realized the complementary nature of identity federation when coupled with an ABAC implementation. Identity federation on its own does not constitute an ABAC solution and an ABAC solution does not rely upon identity federation. Future builds under this project name may or may not include examples of identity federation.
to only those resources residing on Microsoft SharePoint, this solution can be implemented alongside an RBAC implementation, with the lone configuration requirement of enabling attributes inside Microsoft Active Directory or other identity stores as appropriate.

4.2.1.2 Business Policy Language

This build leverages NextLabs technology to decompose natural language business policy into attribute-based digital policies. We implemented example business policies that we feel demonstrate the capabilities of the solution that address business needs. When implementing an ABAC solution, enterprises will need to determine the set of natural language business policies that best meet their access control needs and risk tolerances.

4.2.1.3 Attribute Semantics and Syntax

An ABAC IdAM infrastructure by its intrinsic nature is dependent on a pre-defined set of attribute name:value pairs available for use within its set of rules to determine authorization privileges for users and Web service clients. The use of federation, as with this build, expands the domain of agreed-upon attributes to include trusted federation partners. Often a common attribute dictionary is in use for all parties. However, enterprises may look to a third-party service, typically called a trust broker, to facilitate attribute exchange and normalization.

For the purposes of this build, we have chosen an example set of attribute values that we feel is representative of business needs. When implementing an ABAC solution, enterprises will need to determine the set of attribute syntax and semantics that best meets their unique access control needs.

4.2.1.4 Attribute Provenance

In this build, we utilize Microsoft Active Directory, RSA Adaptive Authentication, and Microsoft SharePoint as sources for attributes. Depending on the types of policy an enterprise wishes to implement in attribute-based logic, there will be diversity in the appropriate sources of attribute information. When planning an ABAC implementation, enterprises should consider their ability to collect the attributes required for access decisions and the level of trust they have with the attribute provider and/or sources of attribute information.

4.2.1.5 Trust Relationships for Identity Federation

The use of identity federation requires a degree of trust between pairs of sharing partners. When establishing this trust relationship, enterprises need to agree upon the technical specification of the trust relationship as well as the types of metadata to be exchanged. Enterprises should make a decision based on their risk profile when determining the stakeholders with which they wish to establish trust relationships.

This build establishes a trust relationship between two theoretical organizations through the exchange of attribute and identity information between two Ping Federate instances using SAML 2.0. In order to demonstrate federation capabilities, this build assumes complete trust between exchanging parties.

4.2.1.6 Human Resources Database/Identity Proofing

This build is based on a simulated environment. Rather than re-create a human resources (HR) database and the entire identity proofing process in our lab, we assume that your organization has the processes, databases, and other components necessary to establish a valid identity.
4.2.1.7 Technical Implementation

The guide is written from a technical perspective. Its foremost purpose is to provide details on how to install, configure, and integrate components. We assume that enterprises have the technical resources to implement all or parts of the build, or have access to companies that can perform the implementation on their behalf.

4.2.1.8 Limited Scalability Testing

We experienced a major constraint in terms of replicating the volume of access requests that might be generated through an enterprise deployment with a sizable user base. We do not identify scalability thresholds in our builds, as those depend on the type and size of the implementation and are particular to the individual enterprise.

4.3 Risk Assessment

According to NIST Special Publication (SP) 800-30-r1, “Risk Management Guide for Information Technology Systems”, “A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence.” The NCCoE recommends that any discussion of risk management, particularly at the enterprise level, begin with a comprehensive review of the Risk Management Framework (RMF) material available to the public. The RMF guidance as a whole proved invaluable in giving us a baseline to assess risks, from which we developed the project, the security characteristics of the build, and this guide.

Using the guidance in NIST’s series of SPs concerning the RMF, the NCCoE worked with IdAM SMEs to enumerate areas of access management risk facing today’s enterprise. We deemed these the tactical risks:

- not implementing or maintaining least privilege for all users
- access rights accumulation violates the separation of duties
- digital identities of external users become orphaned
- authorization policies cannot account for the context of access request

In addition to tactical risk, enterprises face a series of business risks that are influenced by the acquisition, deployment, and maintenance of IdAM systems. We deemed these the strategic risks:

- cost of implementation
- budget expenditure as they relate to investment in security technologies
- compliance with existing industry standards
- risk of alternative or no action
- lack of successful precedents

We translated this risk information to security characteristics. We mapped these characteristics to NIST’s SP 800-53 Rev.4 controls where applicable, as well as other relevant industry and mainstream security standards.
Table 4.4 Security Characteristics and Controls Mapping

Table 1 lists the major use case security characteristics. For each characteristic, the table provides the matching function, category, and subcategory from the NIST Cybersecurity Framework (CSF), as well as mappings to controls from other relevant cybersecurity standards.

Table 4.1 Use Case Security Characteristics Mapped to Relevant Standards and Controls

<table>
<thead>
<tr>
<th>Security Characteristics</th>
<th>CSF Function</th>
<th>CSF Category</th>
<th>CSF Subcategory</th>
<th>NIST SP 800-53 rev4</th>
<th>ISO/IEC 27001</th>
<th>SANS CSC</th>
<th>ISACA COBIT 5</th>
<th>CSA CCMv3.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity and Credentials</td>
<td>Protect</td>
<td>Access Control</td>
<td>PR.AC-1: Identities and credentials are managed for authorized devices and users</td>
<td>AC-1,IA Family</td>
<td>A.9.2.1, A.9.2.2, A.9.2.4, A.9.3.1, A.9.4.2, A.9.4.3</td>
<td>CSC 3-3, CSC 12-1, CSC 12-10, CSC 16-12</td>
<td>DSS05.04, DSS06.03</td>
<td>IAM-02, IAM-03, IAM-04, IAM-08</td>
</tr>
<tr>
<td>Remote Access</td>
<td>Protect</td>
<td>Access Control</td>
<td>PR.AC-3: Remote access is managed</td>
<td>AC-17, AC-19, AC-20</td>
<td>A.6.2.2, A.13.1.1, A.13.2.1</td>
<td>CSC 3-3, CSC 12-1, CSC 12-10, CSC 16-4, CSC 16-12</td>
<td>APO13.01, DSS01.04, DSS05.03</td>
<td>IAM-07, IAM-08</td>
</tr>
<tr>
<td>Access Permissions</td>
<td>Protect</td>
<td>Access Control</td>
<td>PR.AC-4 Access Permissions are managed, incorporating principles of least privilege and separation of duties</td>
<td>AC-2, AC-3, AC-5, AC-6, AC-16</td>
<td>A.6.1.2, A.9.1.2, A.9.2.3, A.9.4.1, A.9.4.4</td>
<td>CSC 3-3, CSC 12-1, CSC 12-10, CSC 16-4, CSC 16-12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4.1 Use Case Security Characteristics Mapped to Relevant Standards and Controls (Continued)

<table>
<thead>
<tr>
<th>Security Characteristics</th>
<th>CSF Function</th>
<th>CSF Category</th>
<th>CSF Subcategory</th>
<th>NIST SP 800-53 rev4&lt;sup&gt;a&lt;/sup&gt;</th>
<th>ISO/IEC 2700&lt;sup&gt;b&lt;/sup&gt;</th>
<th>SANS CSC&lt;sup&gt;c&lt;/sup&gt;</th>
<th>ISACA COBIT 5&lt;sup&gt;d&lt;/sup&gt;</th>
<th>CSA CCMv3.0.1&lt;sup&gt;e&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning</td>
<td>Protect</td>
<td>Information Protection Processes and Procedure</td>
<td>PR.IP-11: Cybersecurity is included in human resources practices (e.g., deprovisioning, personnel screening)</td>
<td>PS Family</td>
<td>A.7.1.1, A.7.3.1, A.8.1.4</td>
<td>APO07.01, APO07.02, APO07.03, APO07.04, APO07.05</td>
<td>IAM-02, IAM-09, IAM-11</td>
<td></td>
</tr>
<tr>
<td>Auditing and Logging</td>
<td>Protect</td>
<td>Protective Technology</td>
<td>PR.PT-1: Audit/log records are determined, documented, implemented, and reviewed in accordance with policy</td>
<td>AU family</td>
<td>A.12.4.1, A.12.4.2, A.12.4.3, A.12.4.4, A.12.7.1</td>
<td>CSC4-2,CSC 12-1, CSC 12-10, CSC 14-2, CSC 14-3</td>
<td>APO11.04</td>
<td>AAC-01</td>
</tr>
<tr>
<td>Access Control</td>
<td>Protect</td>
<td>Protective Technology</td>
<td>PR.PT-3: Access to systems and assets is controlled, incorporating the principle of least functionality</td>
<td>AC-3, CM-7</td>
<td>A.9.1.2</td>
<td>CSC 3-3, CSC 12-1, CSC 12-10, CSC 16-4, CSC 16-12</td>
<td>DSS05.02</td>
<td>IAM-03, IAM-05, IAM-13</td>
</tr>
</tbody>
</table>

---


d. ISACA, “COBIT 5”. [http://www.isaca.org/COBIT/Pages/Product-Family.aspx](http://www.isaca.org/COBIT/Pages/Product-Family.aspx)

### 4.5 Technologies

Table 4.2 provides a breakout of the contents of table 4.1 organized by the products used within this build. This breakout shows the security controls coverage that each product supports.

**Table 4.2 Use Case Security Characteristics Mapped to Relevant Build Products**

<table>
<thead>
<tr>
<th>Security Characteristics</th>
<th>Product(s)</th>
<th>CSF Subcategory</th>
<th>NIST SP 800-53r4</th>
<th>ISO/IEC 27001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Access</td>
<td>Microsoft SharePoint, NextLabs Policy Controller and Control Center, Ping Federate RP, Ping Federate IdP</td>
<td>PR.AC-3: Remote access is managed</td>
<td>AC-17, AC-19, AC-20</td>
<td>A.6.2.2, A.13.1.1, A.13.2.1</td>
</tr>
<tr>
<td>Provisioning</td>
<td>Microsoft Active Directory</td>
<td>PR.IP-11: Cybersecurity is included in human resources practices (e.g., deprovisioning, personnel screening)</td>
<td>PS Family</td>
<td>A.7.1.1, A.7.3.1, A.8.1.4</td>
</tr>
</tbody>
</table>
This build implements the security characteristics through available products, described below, from NCEP organizations. Section 5, Architecture, provides additional insight into the way we used the products.

- The build is centered on a resource server to be protected by the ABAC solution. In this case, Microsoft SharePoint was used. It is a web-based application within the Windows operating environment commonly, SharePoint is deployed as a document management system for intranet, extranet, or cloud repository purposes. SharePoint natively uses an RBAC authorization environment, but it also supports the use of attributes within the user transaction request, a capability Microsoft refers to as being “claims aware.” SharePoint also allows for tagging data within its repository, which can be leveraged as object attributes.

- Another important component of the build is identity management software, in this case, Microsoft Active Directory (AD). AD is a set of services that reside within the Windows server environment. AD functions as an identity repository based on LDAP technology, but also provides authentication and authorization services. AD also includes the ability to provision and de-provision user identities and the creation, modification, and deletion of subject attributes.

- The build needed PEP functionality. It is provided by NextLabs Entitlement Management, which interfaces and integrates with products like SharePoint and SAP to provide finer granularity of access decisions than that available using the native access control mechanisms. Entitlement Management is closely coupled with the target application. It traps user access requests and passes access decisions to the policy decision point (PDP).

- Policy lifecycle management and auditing/reporting are facilitated by the NextLabs Control Center, which hosts policy administration point (PAP) functionality, where attribute-based policies are defined and deployed. The NextLabs Policy Controller, as an element of Control Center, hosts the PDP, which uses the policy definitions and subject, object, and environmental attributes to make an access accept-or-deny decision that the PEP enforces. Control Center also includes dashboards, analytics, reports, and monitoring to offer insight into access patterns.

### Table 4.2 Use Case Security Characteristics Mapped to Relevant Build Products

<table>
<thead>
<tr>
<th>Security Characteristics</th>
<th>Product(s)</th>
<th>CSF Subcategory</th>
<th>NIST SP 800-53r4</th>
<th>ISO/IEC 27001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auditing and Logging</td>
<td>Microsoft SharePoint, NextLabs Policy Controller, Ping Federate RP, Ping Federate IdP, RSA Adaptive Authentication</td>
<td>PR.PT-1: Audit/log records are determined, documented, implemented, and reviewed in accordance with policy</td>
<td>AU family</td>
<td>A.12.4.1, A.12.4.2, A.12.4.3, A.12.4.4, A.12.7.1</td>
</tr>
<tr>
<td>Access Control</td>
<td>NextLabs Policy Controller and Entitlement Manager and Control Center</td>
<td>PR.PT-3: Access to systems and assets is controlled, incorporating the principle of least functionality</td>
<td>AC-3, CM-7</td>
<td>A.9.1.2</td>
</tr>
</tbody>
</table>
The build includes a federation server/platform for exchanging identities and attributes. Ping Identity’s PingFederate serves as a federation identity system or trust broker, an identity management component, and supports integrated single-sign-on (SSO) within an enterprise IdAM infrastructure. It supports standards-based protocols such as SAML, OAuth, and OpenID Connect. Its trust broker capabilities allow for necessary transformation and interface options between federated partners and internal proprietary target resources. When used within an identity provider, it offers options for integrating with authoritative attribute sources.

The build has an authentication server that supports multifactor authentication. For this build, RSA Adaptive Authentication (AA), which is an authentication and environmental analysis system, provides this functionality. Its capabilities include a variety of adaptive opportunities, such as SMS texting, fingerprint analysis, and knowledge-based authentication. From an environmental perspective, AA collects information such as patch level, operating system, and location, and generates a risk score associated with user authentication. A risk score threshold can then be defined, which, if exceeded, can force a user to step up to an additional authentication mechanism.

A final necessary component of the build is a certificate authority. In this case Symantec’s Managed PKI Service product is used for secure issuance of PKI-based certificates. The Symantec certificates enable mutual transport layer security (TLS), digital signatures, and any explicit encryption that is in use outside of TLS, such as for data-at-rest within an IT environment.
# 5 Architecture

1. Overview ....................................................................................................................................................................................... 26
2. ABAC Architecture Considerations .............................................................................................................................................. 27
3. Security Characteristics ............................................................................................................................................................... 38
4. Features and Benefits ................................................................................................................................................................. 39
5.1 Overview

The following sections detail the ABAC and identity federation architecture that NCCoE staff members and collaborators built. The architecture description details how components from five NCEPs were integrated to achieve the following demonstrable capabilities:

5.1.1 User Authentication and the Creation of an Authentication Context

Our scenario starts with an unauthenticated user attempting to access a target resource for the first time. The user’s browser is redirected to his or her home organization (the IdP) for authentication and includes, as required for the target resource, additional (step-up) authentication, and gathering of environmental attributes and authentication context information about the user.

5.1.2 Federation of a User Identity and Attributes

This build demonstrates the federation of subject and environmental attributes between an IdP and an RP. This means that, after the user is authenticated by his or her IdP, the federation protocol that initially redirected the user to the IdP is now used to redirect the user back to the RP carrying the requested identity and attribute information.

5.1.3 Fine-Grained Access Control through a PEP Closely Coupled with the Application

Out of the box, SharePoint access control is more oriented to role-based or group-based Discretionary Access Control (DAC). In this build, we enhance the SharePoint access control environment through the deployment of a closely integrated policy enforcement allowing for a finer degree of granularity based on subject, object, and environmental attributes.

5.1.4 The Creation of Attribute-Based Policy Definitions

This build allows for the translation of business policies into a set of attribute-based policy definitions. These policy definitions establish a relationship between subject, object, and environmental attributes that controls a user’s ability to access the RP’s resources.

5.1.5 Secondary Attribute Requests

This build provides the ability to make runtime requests for additional attributes from the IdP, should insufficient attributes be presented when making an access decision. When a user accesses a particular

---

1. This project has the overarching goal of demonstrating technical implementations of standards-based ABAC functionality. In enumerating technology relevant to this effort, we worked closely with experts from the identity and access management community. During those discussions, we realized the complementary nature of identity federation when coupled with an ABAC implementation. Identity federation on its own does not constitute an ABAC solution and an ABAC solution does not rely upon identity federation. Future builds under this project name may or may not include examples of identity federation.
Chapter 5. Architecture

resource, or returns to access additional resources, the access control components that we have associated with SharePoint might find that additional subject attributes are needed beyond those that were initially provided. Our build includes components able to search a local cache for the missing attributes and if not there, issue a new request to the IdP via a SAML attribute request/response for the missing user attributes.

5.1.6 Allow RP Access Decisions on External Identities without the Need for Pre-Provisioning

This build relies upon the trust relationship between the IdP and RP, which enables identity and attribute federation. Once this trust relationship has been established between two organizations, the relying party is afforded the ability to make run-time access decisions on any individual presenting a credential from the IdP without the need to pre-provision that individual.

5.2 ABAC Architecture Considerations

There are many facets to architecting an ABAC system. As noted in section 4.2.1, Assumptions, these include the development of policy, procedure, and/or functional requirements before the selection of technology components. Organizations wishing to implement an ABAC system should conduct robust requirements engineering, taking into consideration the operational needs of each system stakeholder. Standards such as ISO/IEC 15288:2015, Systems and software engineering - System life cycle processes\(^1\) and NIST SP 800-160, Systems Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems\(^2\) provide guidance in this endeavor.

From a technical perspective, this section outlines a few of the options that an architect will face, and section 5.2.6, Architecture Diagram and Components, presents the actual architecture chosen for this build.

5.2.1 Industry Standards

When selecting ABAC technologies, it is important to consider the protocols implemented by each technology and whether those protocols are defined by a standards organization. Utilizing standard protocols promotes product interoperability and modularity, and may offer standardized APIs in the event that system requirements drive the need for custom components.

As mentioned earlier, one of the standards for implementing ABAC is XACML. Built on top of XML, XACML offers a core set of rule capabilities for making attribute-based policy definitions and also specific request and response messages for exchange between PEPs and PDPs. Specific details of the XACML 3.0 architecture can be found in the OASIS documentation.\(^3\)

---

\(^1\)http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63711


Although XACML was developed primarily to fill the need for a standard ABAC protocol, other standard
protocols and architectures may be relevant to ABAC use cases. Next Generation Access Control\textsuperscript{1},
developed by the International Committee for Information Technology Standards, outlines an access
control architecture that supports the use of attributes. OAuth 2.0\textsuperscript{2}, ratified by the Internet Engineering
Task Force (IETF), serves as a rights delegation protocol that grants access to protected resources by
defining the allowable user actions for those resources referred to as “scopes.”

When system requirements include identity federation, protocols such as SAML 2.0 and OpenID Connect
can define the syntax and semantics for passing identity and attribute information across organization
bounds.

5.2.2 PEP Placement

As it is in the XACML architecture, the PEP is a very important ABAC component since it enforces the
actual access control decision. The location of the PEP may affect the types of access requests the ABAC
system is able to trap and send to the PDP for decisions. It may also contribute to how efficiently the
system handles large numbers of access requests. Common options for PEP placement include:

- closely coupling it within a software program
- using an agent to front-end a web browser-based application
- placing it at an enterprise gateway position in order to ABAC-enable a set of applications

The PEP may also be asked to perform additional functions that require a specific PEP placement. Under
the XACML standard, the PEP can be configured to handle “out-of-band” instructions known as
obligations (mandatory directives) and advice (optional). These instructions trigger secondary actions in
addition to the access decision enforcement. An example of an obligation would be where a person was
allowed access to a target resource, but the PEP is directed to initiate a royalty payment for its use.

5.2.3 PDP Distribution

The PDP operates a rule-based engine that is called upon to adjudicate access permissions to a selected
resource. Typical ABAC installations get involved in deciding whether to locate PDPs centrally where each
PDP supports multiple PEPs, to dedicate one PDP to each PEP, or to pursue a hybrid of the two
approaches. Different PDP distributions can be associated with various performance and latency
characteristics.

\textsuperscript{1}INCITS, INCITS 499-2013, \textit{Information Technology - Next Generation Access Control -
Functional Architecture (NGAC-FA)}.
tools.ietf.org/html/rfc6749
5.2.4 Multi-Vendor

ABAC systems have traditionally been classified as proprietary or standards based. Those that are standards based give the option of mixing and matching among system components rather than requiring all components to come from the same vendor. A multi-vendor-implementation solution sometimes needs some advance investigation to ensure that the standardized components will work together as well as promised.

5.2.5 Caching

There are several locations in an ABAC system implementation for an architect to consider the use of memory caching to improve performance. Considerations include caching decisions at the PEP, rules at the PDP, and user attributes at the RP.

Section 4.5 provides an overview of the technologies used in this architecture, while Section 5.1 details the functionality found in this build. This section documents how each of the technologies in this build interoperate to achieve the build’s functionality. Individuals interested in how these components were installed, configured, or integrated should consult Volume C How-To Guides of this publication.

5.2.6 Architecture Diagram and Components

Figure 5.1 illustrates the logical interactions of the components in this build. Interactions are broken down into browser-based or non-browser-based communications. All components in this build are either commercially available through the applicable vendor or can be found publicly with the release of this practice guide.
Figure 5.1 ABAC Build 1 Architecture
The components in figure 5.1, which were available products from NCEP organizations that met the build’s functional requirements, provide the following capabilities to this build:

- **Microsoft AD** acts as a user identity management repository for the IdP. This includes the ability to provision and de-provision user identities; the creation, modification, and deletion of subject attributes; and the provisioning and de-provisioning of subject attributes to specific user identities. In this build, AD is the only source for subject attributes.

- **RSA AA** gathers environmental information about the user and the user's system or agent at the time of authentication. AA collects information such as patch level, operating system, and location, and it generates a risk score associated with the user authentication. A risk score threshold can then be defined in AA, which, if exceeded, can force a user to step up to one of the additional authentication mechanisms. In this build, information collected by AA to generate a risk score is also passed through PingFederate-IdP to the RP side of the operation to be used as environmental attributes.

- The **RSA AA event log** contains the transaction ID of each user authentication and the associated environmental information collected by RSA AA at the time of authentication.

- **Ping Identity PingFederate-IdP** serves as a federation system or trust broker for the IdP. PingFederate-IdP provides initial user authentication and retrieval of user attributes to satisfy SAML requests from the RP. Once the user has been authenticated, PingFederate-IdP queries subject attributes from AD and environmental attributes from the RSA AA event log. PingFederate-IdP packages both subject and environmental attributes in a SAML 2.0 token to be sent to the RP.

- The **SCE Plugin** is an RSA component that handles communications between the PingFederate-IdP and the RSA AA. It is responsible for passing the RSA AA transaction ID for the user authentication that PingFederate-IdP uses to query the RSA AA event log.

- **Ping Identity PingFederate-RP** serves as the trust broker for SharePoint. When the user requires authentication, PingFederate-RP redirects the user to the IdP via a SAML request to get the necessary assertions. Once authenticated, PingFederate-RP arranges for the browser's HTTPS content to have the proper information in proper format for acceptance at the target resource (SharePoint). PingFederate-RP has the option to utilize the Apache Directory Server as a just-in-time (JIT) cache. Secondary attribute requests can also be made by PingFederate-RP via a SAML query initiated by the PIP Plugin and the Protocol Broker.

- Microsoft SharePoint serves as a typical enterprise repository and in this build, it stores the target resources that users wish to access. SharePoint natively uses an RBAC authorization environment, but it also supports the use of attributes, a capability Microsoft refers to as “claims aware.” SharePoint accepts assertions from PingFederate-RP and stores asserted attributes as claims. SharePoint also allows for the tagging of data within its repository, which can then be leveraged as object attributes.

- **Microsoft SharePoint Security Token Handler** resides inside of SharePoint, validating the token sent by PingFederate-RP.

- **Microsoft SharePoint Claims Principal** is the object inside of SharePoint where attribute assertions are stored as claims.

- NextLabs Entitlement Management is closely coupled with SharePoint. It performs the PEP functionality, trapping user access requests. As the PEP, Entitlement Management is responsible for gathering object attributes from SharePoint and subject and environmental attributes from the claims principal at the time of the access request. Entitlement management then passes this information in the form of an access decision request to the NextLabs Policy Controller.
NextLabs Policy Controller is a component of the NextLabs Control Center that is closely coupled with the SharePoint instance. The Policy Controller is responsible for providing PDP capabilities. The Policy Controller receives attribute-based policies from the Control Center and uses these policies to respond to access requests from Entitlement Management.

NextLabs Control Center serves as the PAP, where attribute-based policies are created, updated, and deployed using a built-in graphical user interface (GUI). The Control Center also provides auditing, logging, and reporting functions for the SharePoint access requests and decisions.

PIP Plugin is a software extension of NextLabs Policy Controller that enables it to acquire unavailable attributes required for policy evaluation at run time from RP or IdP by communicating with Protocol Broker on an HTTPS channel protected by mutual TLS.

Protocol Broker is a Web application that retrieves attribute values by accepting attributes to be queried from the NextLabs Plugin and querying the PingFederate-RP by issuing a SAML 2.0 Assertion Query/Request.

The Custom Data Store is a plugin built using PING SDK that enables the RP to query the IdP and provides the resulting attribute value back to the Ping Federate RP.

The Apache Directory Server is an LDAP version 3-compliant directory server developed by the Apache Software Foundation that works as a JIT cache for PingFederate-RP. It stores subject attributes and other relevant information from the SAML 2.0 response that an RP receives from an IdP.

Symantec Trust Center Account for Enterprise is used for secure issuance of PKI-based certificates throughout this build. The Symantec certificates enable mutual TLS, digital signatures, and any explicit encryption that is in use outside of TLS, such as for data-at-rest in the RP’s JIT cache.

5.2.7 UML Diagram

The architecture shown in figure 5.1 can, in practice, support different types of sequential operations. We have chosen to initially implement, demonstrate, and document two generic types of sequential ABAC operations as being representative of the core operations of the architecture. Figure 5.2 contains a ladder diagram that represents the initial flow of the ABAC architecture, where an unauthenticated user tries to access a resource on SharePoint.
Figure 5.2 UML Sequence Diagram
The sequence starts in the top of figure 5.2 when a user browses to, and attempts to access, a protected resource in SharePoint.

1. SharePoint inspects the user’s HTTP content and finds that the user has not been previously logged in (i.e., not authenticated), and therefore re-directs the browser to PingFederate-RP via use of the WS-Federation protocol.

2. The WS-Federation request is interpreted by PingFederate-RP as a request for authentication and for attributes, and the user is redirected to PingFederate-IdP carrying a SAML authentication request and SAML attribute request.

3. PingFederate-IdP does an initial (single factor) authentication of the user, and, if successful, receives the requested subject attributes.

4. PingFederate-IdP then redirects the user’s browser to RSA AA to enhance the initial authentication.

   **Note:** In practice this secondary authentication can be conditionally done based upon the type of protected resource for which access is requested or upon other conditions such as environment. The current installation always calls for the second level of authentication to demonstrate what is known as multi-factor authentication (MFA), and for this build achieves it via sending an SMS text message and expecting a particular response. The RSA AA product has additional options that are not being demonstrated at this time.

5. Upon successful completion of the MFA operation, the user is redirected back to PingFederate-IdP. At this time, PingFederate-IdP can query the RSA AA event log for environmental attributes that add context to the authentication.

6. PingFederate-IdP issues a SAML 2.0 token containing the user’s identity and attribute information, and redirects the user’s browser to PingFederate-RP.

7. PingFederate-RP accepts the SAML 2.0 response and issues a WS-Federation response back to SharePoint with the HTTP carrying the authentication and attribute information.

   At this point the user’s browser is issued a “FedAuth” cookie, establishing a session with SharePoint, and resides there until the session is terminated. The rest of this flow occurs as communications internal to the RP or as web service calls back to the IdP, unknown to the user. Once this session is established, the system is configured to allow the NextLabs components to handle access requests to SharePoint. After the WS-Federation response, the subject and environmental attributes from the IdP are stored in the SharePoint Claims Principal.

8. Access requests by the authenticated user are now trapped by the NextLabs Entitlement Management PEP, which gathers the subject and environmental attributes stored in the Claims Principal and the object attributes stored in SharePoint, and submits the access request to the Policy Controller PDP for adjudication.

9. The Policy Controller uses the attributes provided by the PEP and the policy established by the Control Center to determine an access allow or deny. If the PDP is not presented with enough attributes to make an access decision, it has the option of initiating a secondary attribute query, which is detailed in Figure 3 and discussed later.

10. Once an access decision has been made, the Policy Controller responds back to the Entitlement Management PEP, which enforces the decision.

**Figure 5.3** contains a ladder diagram that represents a flow of this ABAC architecture where an authenticated user tries to access a resource on SharePoint but there is a need to initiate a secondary attribute request. If needed, this flow is initiated by the NextLabs Policy Controller in Step 9.
Figure 5.3  Secondary Attribute Request Flow
The basic steps of the figure 5.3 flow:

1. When the policy controller does not receive the attributes required to make a decision, a secondary attribute request will be initiated by calling the PIP Plugin.

2. PIP Plugin is a registered plugin with the NextLabs Policy Controller. It implements the interface dictated by the NextLabs software. By virtue of this implementation it receives the subject and name of the attribute that is required for the policy decision.

3. When the subject and attribute name are received, the PIP Plugin checks its local short-term cache (in this build, configured to hold values for two seconds) to see if the needed attribute for the subject was recently requested.

4. If the attribute is still in cache, the value is returned to the Policy Controller. If the value is not in cache, the PIP Plugin initiates an HTTPS request to the Protocol Broker.

5. The Protocol Broker receives the attribute name and subject from the HTTPS request and forwards them as a signed SAML 2.0 Attribute Query to PingFederate-RP on a channel protected by mutual TLS.

6. Once PingFederate-RP receives the SAML 2.0 attribute query, it sends an LDAP request to the JIT cache to see if the attribute was previously queried in a secondary request.

7. If the subject does not have the attribute value assigned in the JIT cache, PingFederate-RP will forward the subject and attribute name to the Custom Data Store plugin. The Custom Data Store plugin acts as a pointer back to the PingFederate-IdP. To do this, the Custom Data Store dispatches an HTTPS request to the PingFederate-RP with the PingFederate-IdP as the attribute query point.

8. PingFederate uses an HTTPS query to form a SAML 2.0 attribute query and dispatch it to the PingFederate at the IdP.

9. The PingFederate at the IdP accepts the SAML 2.0 request, verifies if the user has the attribute of need, and replies back to the PingFederate-RP with a SAML 2.0 response.

10. PingFederate-RP validates the SAML 2.0 response, retrieves attribute values, and responds to the original Custom Data Store HTTP request with the attribute values.

11. The Custom Data Store then responds to the PingFederate-RP attribute request with an attribute response.

12. The PingFederate-RP constructs a SAML 2.0 response and sends it to the Protocol Broker.

13. The Protocol Broker retrieves the attribute or exception from the SAML 2.0 response and forwards it to the NextLabs plugin, which in turn passes the attribute or exception back to the Policy Controller.

5.2.8 NCCoE Design Considerations

Section 5.2, ABAC Architecture Considerations, outlined the architectural topics and options that entered into our decision making for this first ABAC build and demonstration. Now that the chosen ABAC functionality has been described and the flow and sequencing explained, in this sub-section we summarize the architectural directions that were chosen for this particular build, and why.

5.2.8.1 Industry Standards

The use of XACML and its importance to ABAC functionality was introduced in section 5.2.6. Its core parts are the request/response protocol between PEP and PDP, the rule language, and the use of obligation and advice that the PDP can forward to the PEP. Use of a standard like XACML gives an IdAM infrastructure
implementation potential cost saving as heterogeneous interchangeability of operational components can be more easily implemented.

The use of SAML 2.0 provided advantages from several perspectives. From its documented set of approved federation profiles, the Web Browser SSO Profile (referred to here as “Web SSO”) has a large following in the industry and was chosen for the browser interface because its authentication sequencing stepped between PingFederate-RP, PingFederate-IdP, and the RSA AA system.

SAML 2.0 core was used within the SAML Web SSO exchange, but was also used as a standalone for its request/response protocol for backend attribute exchanges of NextLabs’ PIP Plugin to and from PingFederate-RP (via the Protocol Broker), and for back-end attribute exchanges from PingFederate-IdP to PingFederate-RP.

WS-Federation is a federation protocol that spans important federation functionality, ranging from authentication to metadata, support for pseudonyms, and more. Our use is limited but still key: to carry an authentication request from SharePoint to PingFederate-RP, and then to handle the return response with its identity and user attribute information.

LDAPS, the TLS version of the LDAP standard for interfacing to directory stores, is used in two places in this build. One is PingFederate-RP to its JIT cache based on Apache Directory Server, and the other is PingFederate-IdP to the Microsoft AD LDAP store. Other standards in use include PKI for the structure of the server certificates that are in use, and within TLS operational algorithms. TLS itself is an important standard for promoting communications confidentiality and integrity.

5.2.8.2 PEP Placement

There is a single PEP in this ABAC build with the purpose of controlling the operations of the SharePoint authorization functionality at a finer level of granularity than is available with the RBAC-oriented access control that comes with SharePoint out of the box. The NextLabs Entitlement Management PEP product was chosen due to meeting our requirements, and by its nature it is integrated with and closely coupled with SharePoint. The NextLabs PEP can be considered to be co-located with the SharePoint protected resource.

5.2.8.3 PDP Distribution

With only one PEP in this build, the decisions on PDP quantity and location(s) for placement were simpler than one would find in a typical enterprise installation. The NextLabs Policy Controller PDP is co-located with SharePoint and the PEP.

5.2.8.4 Multi-Vendor

The ABAC implementation represented in this build is a heterogeneous set of IdAM components that have been successfully integrated to achieve the system objectives. To accomplish this we worked closely with our NCEP collaborator in order to design an interoperable architecture. Each component performed its functions as required, and Volume C of this guide describes the set of NCCoE experiences and supplemental functionality that was incorporated to achieve the functional objectives.

5.2.8.5 Caching

Caching is a common topic in system integration work as architects work to achieve efficiencies required for their particular functionality. In the current build, two caches have been explicitly implemented by the NCCoE development team:
NextLabs PIP Plugin contains a local cache, developed using the EhCache library. This cache stores attributes for 2 seconds and adds efficiency to the system should multiple requests for the same subject and attribute value pairing occur in quick succession (with 2 seconds).

A JIT cache was developed for PingFederate-RP, using Apache Directory Server. It is used to cache user attributes that are retrieved by PingFederate-RP for a finite time (such as up to 24 hours) to avoid future repeated secondary attribute calls to the IdP.

5.3 Security Characteristics

In this section we re-introduce the security characteristics and security controls that were first introduced in Sections 4.4 and 4.5, and relate each here to the NCEP partner products that are being used in this ABAC build.

Identity and Credentials and Their Use for Authorized Devices. In NIST SP 800-53 this is tied to AC-1, and in the NIST Cybersecurity Framework to PR.AC-1: “Identities and credentials are managed for authorized devices and users.” In this build, both user and system identities are managed to ensure linkage with these security controls. Where applicable systems are given PKI-based credentials for use with TLS via the Symantec Managed PKI Service. User authentication in this first build is MFA with one factor being name and password via PingFederate-IdP and AD, while the second is an SMS text message sent to a cellular device conducted by the RSA AA. The RSA AA system offers other options for use as the second factor of authentication through its multi-credential framework.

Remote Access Being Managed. Several of the NCEP products are involved in ensuring efficient and secure remote access. The two Ping Identity PingFederate installations have federation and authentication features that allow the RP to accept external identities for remote access. SharePoint via WS-Federation trusts external identities sent from PingFederate. NextLabs products enable ABAC functionality for SharePoint access decisions and allow for the auditing and logging of access requests.

Access Permissions. ABAC systems manage access permissions by defining attribute-based rules that specify what subject attributes are needed to access resources with a given set of object attributes, under a set of environmental conditions. In this build, this functionality is handled by NextLabs products. A NextLabs Control Center allows for creation of attribute-based policies and makes access decisions based on those policies via its Policy Controller.

Encryption and Digital Signature. Browser-based communications with SharePoint are HTTPS-based, and LDAP is used for all interfacing with AD. All system endpoints are equipped with PKI certificates issued by the Symantec Managed PKI Service, and TLS is in use for system-level point-to-point transactions. Examples include full encryption of SAML request/response transactions such as between PingFederate-RP and PingFederate-IdP.

Provisioning. Identities are provisioned, stored, and de-provisioned inside of AD. This process occurs manually through the native Microsoft Windows Server GUI. AD also handles the assigning of subject attributes to specific user identities.

Object attributes are provisioned via SharePoint. SharePoint sites or individual files can be “tagged” with object attributes by adding columns to the SharePoint site table or document library. The titles of these columns serve as attribute names and the content of the columns serves as the values of attributes for the specific object.

Auditing and Logging. Each product in this build supports a logging mechanism detailing activities occurring within that component. Access requests can be audited using the NextLabs Reporter, where the user, access decision, and policy enforced can be viewed for each access request.
Access Control. Fundamentally, this build enhances the native RBAC capabilities of SharePoint by adding ABAC functionality. This is achieved through the NextLabs Entitlement Management PEP, which traps access requests, and the Policy Controller PDP, which makes access decisions using attribute-based policies. Organizations implement the concept of least privilege by defining attribute-based policies in the NextLabs Control Center and assigning applicable attributes to subjects and objects using AD and SharePoint. A wider range of access control decisions is enabled through the use of environmental attributes, which can be obtained from RSA AA in this build.

5.4 Features and Benefits

This section details some of an ABAC system’s potential benefits through risk reductions, cost savings, or access management efficiencies. As with any reference architecture, the exact benefits derived will be dependent on the organization’s individual implementation requirements and the scenarios to which an organization wishes to apply an ABAC model.

5.4.1 Support Organizations with a Diverse Set of Users and Access Needs

RBAC meets practical limits as roles and their associated access requirements grow in diversity and complexity. This often leads to the overloading of access privileges under a single role, the assignment of multiple roles to a single user, or the escalation of the number of roles the enterprise needs to manage. Moving to an ABAC model allows organizations to specify policy based on a single attribute or a combination of attributes that represents the specific access needed by an individual. This helps eliminate the potential for privilege creep.

5.4.2 Reduce the Number of Identities Managed by the Enterprise

When organizations wish to provide access to users from external security domains, they have the option to provision local identities for these external users. These identities must then be managed by the enterprise. This scenario incurs the costs associated with these management efforts and also presents risk to the enterprise because these accounts could be orphaned as the users’ access privilege requirements change at their home organization. Identity federation can address these issues by allowing organizations to accept digital identities from external security domains, but leave the management of these identities to the users’ home organization.

5.4.3 Enable a Wider Range of Risk Decisions

The ability to define attribute-based policies affords organizations the extensibility to implement a wider range of risk decisions in access control policy than otherwise would be available under an RBAC system. Specifically, the ability to leverage environmental attributes allows for the inclusion of relevant context such as location of access, time of day, threat level, and client patch level into automated decision logic.

5.4.4 Support Business Collaboration

ABAC combined with identity federation helps reduce barriers to sharing resources and services with partner organizations. Under the ABAC model, a partner’s user identities and appropriate access policies
for those identities do not need to be pre-provisioned by the RP. Instead, access decisions can be made on partner identities using attributes provided by the partner.

5.4.5 Centralize Auditing and Access Policy Management

ABAC can improve the efficiency of access management by eliminating the need for multiple, independent, system-specific access management processes, replacing them with a centralized PDP and PAP. In this way access decisions across multiple applications could be audited centrally at the PDP, while policies could be created and deployed centrally at the PAP, but enforced locally via an application-specific PEP. The ability to externalize and centrally manage access policies may also simplify compliance processes by reducing the number of places that need to be audited.
Chapter 6. Future Build Considerations

6 Future Build Considerations

6.1 Potential Additions to This Build ................................................................. 42
6.2 Future Builds ............................................................................................... 42
6.1 Potential Additions to This Build

To help us expand this work in future builds, we need feedback from the user community to prioritize additional capabilities and learn from the identity and access management vendor community which commercial products provide those capabilities.

Here are some of the potential technical capabilities that may be added to this build:

- Demonstration of a wider array of authentication methods including but not limited to smart card, biometric and OTP tokens.
- The ability to support RP-initiated step up authentication. After the user has already authenticated, allow the RP to force the user to undergo advanced authentication based on the object they are accessing.
- More robust logic relative to the current WS-Federate flow. Potential replacement of or supplement to the existing use of a WS-Federation request to limit the need to have a canned set of attributes with the initial user authentication, and to allow for attributes to be acquired on demand in any subsequent browser-based queries.
- Additional environmental attributes. Any potentially interesting sources for environmental attributes that may be useful for decisions based on risk.
- Implementation of SCIM 2.0 for cross-domain identity and attribute management.
- Expand the implementation to include multiple IdP sources. As part of this implementation, at least one home administrative realm discovery approach based on available standards-based methods.
- Pursue an alternate federation approach such as OpenID Connect, an alternative to SAML-based federation that supports the types of browser-based queries in our scenario.
- Expand the set of protected resources beyond the single-product instance of SharePoint.

6.2 Future Builds

In addition to potential updates and add-ons to this first build, there is potential for the development and implementation of new ABAC architectures under this build. To explore these various architectures, the NCCoE would like to engage with any individual or company with commercially or publicly available technology relevant to the ABAC model. The NCCoE recently published a Federal Register notice (https://federalregister.gov/a/2015-20041) inviting parties to submit a letter of interest to express their desire and ability to contribute to this effort. Interested parties will enter into a consortium Cooperative Research and Development Agreement with NIST and anticipate publishing federal register notice.

Some topics of interest for future builds include:

- Use of other protocols that may be relevant to the ABAC model such as OAuth, OpenID Connect, and User Managed Access.
- Demonstration additional options for PDP and PEP placement, such as a loose coupling with the application.
- Potential architectures that use the ABAC model to protect cloud applications to include software as a service (SaaS) applications.
- Integration of the ABAC model with physical access control systems.
- Integration of the ABAC model with legacy technology where PEP integration is not feasible.

All interested parties are encouraged to engage the NCCoE with additional ideas and system requirements by reaching out to abac-nccoe@nist.gov.
# Appendix A  Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>Adaptive Authentication</td>
</tr>
<tr>
<td>ABAC</td>
<td>Attribute Based Access Control</td>
</tr>
<tr>
<td>AC</td>
<td>Access Control</td>
</tr>
<tr>
<td>AD</td>
<td>Microsoft Active Directory</td>
</tr>
<tr>
<td>CSA</td>
<td>Cloud Security Alliance</td>
</tr>
<tr>
<td>CSF</td>
<td>Cybersecurity Framework</td>
</tr>
<tr>
<td>DAC</td>
<td>Discretionary Access Control</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>HTTP Secure</td>
</tr>
<tr>
<td>IdAM</td>
<td>Identity and Access Management</td>
</tr>
<tr>
<td>IdP</td>
<td>Identity Provider</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IPsec</td>
<td>Internet Protocol Security</td>
</tr>
<tr>
<td>ISACA</td>
<td>Information Systems Audit and Control Association</td>
</tr>
<tr>
<td>ISO/IEC</td>
<td>International Organization for Standardization/International Electrotechnical Commission</td>
</tr>
<tr>
<td>JIT</td>
<td>just-in-time</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>MFA</td>
<td>Multi-Factor Authentication</td>
</tr>
<tr>
<td>NCCoE</td>
<td>National Cybersecurity Center of Excellence</td>
</tr>
<tr>
<td>NCEP</td>
<td>National Cybersecurity Excellence Partner</td>
</tr>
<tr>
<td>NGAC</td>
<td>Next Generation Access Control</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>OAuth</td>
<td>Open Standard for Authorization</td>
</tr>
<tr>
<td>OIDC</td>
<td>OpenID Connect Core</td>
</tr>
<tr>
<td>PAP</td>
<td>Policy Administration Point</td>
</tr>
<tr>
<td>PDP</td>
<td>Policy Decision Point</td>
</tr>
<tr>
<td>PEP</td>
<td>Policy Enforcement Point</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>RBAC</td>
<td>Role Based Access Control</td>
</tr>
<tr>
<td>RP</td>
<td>Relying Party</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>34</td>
<td>SAML</td>
</tr>
<tr>
<td>---</td>
<td>------------</td>
</tr>
<tr>
<td>35</td>
<td>SAP</td>
</tr>
<tr>
<td>36</td>
<td>SCI</td>
</tr>
<tr>
<td>37</td>
<td>SMS</td>
</tr>
<tr>
<td>38</td>
<td>SP</td>
</tr>
<tr>
<td>39</td>
<td>SP</td>
</tr>
<tr>
<td>40</td>
<td>SSO</td>
</tr>
<tr>
<td>41</td>
<td>TLS</td>
</tr>
<tr>
<td>42</td>
<td>URL</td>
</tr>
<tr>
<td>43</td>
<td>WS-Federation</td>
</tr>
<tr>
<td>44</td>
<td>XACML</td>
</tr>
<tr>
<td>45</td>
<td>XML</td>
</tr>
</tbody>
</table>
DISCLAIMER

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST or NCCoE, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Special Publication 1800-3c
CODEN: NSPUE2

Organizations are encouraged to review all draft publications during public comment periods and provide feedback. All publications from NIST’s National Cybersecurity Center of Excellence are available at http://nccoe.nist.gov.

Comments on this publication may be submitted to: abac-nccoe@nist.gov

Public comment period: September 30, 2016 through December 4, 2016

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
9600 Gudelsky Drive (Mail Stop 2002) Rockville, MD 20850
Email: abac-nccoe@nist.gov
NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE) at the National Institute of Standards and Technology (NIST) addresses businesses’ most pressing cybersecurity problems with practical, standards-based solutions using commercially available technologies. The NCCoE collaborates with industry, academic, and government experts to build modular, open, end-to-end reference designs that are broadly applicable and repeatable. The center’s work results in publicly available NIST Cybersecurity Practice Guides, Special Publication Series 1800, that provide users with the materials lists, configuration files, and other information they need to adopt a similar approach.

To learn more about the NCCoE, visit http://nccoe.nist.gov. To learn more about NIST, visit http://www.nist.gov.

NIST CYBERSECURITY PRACTICE GUIDES

NIST Cybersecurity Practice Guides (Special Publication Series 1800) target specific cybersecurity challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the adoption of standards-based approaches to cybersecurity. They show members of the information security community how to implement example solutions that help them align more easily with relevant standards and best practices.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. The documents in this series do not describe regulations or mandatory practices, nor do they carry statutory authority.

ABSTRACT

Enterprises rely upon strong access control mechanisms to ensure that corporate resources (e.g., applications, networks, systems and data) are not exposed to anyone other than an authorized user. As business requirements change, enterprises need highly flexible access control mechanisms that can adapt. The application of attribute based policy definitions enables enterprises to accommodate a diverse set of business cases. This NCCoE practice guide details a collaborative effort between the NCCoE and technology providers to demonstrate a standards-based approach to attribute based access control (ABAC).

This guide discusses potential security risks facing organizations, benefits that may result from the implementation of an ABAC system and the approach that the NCCoE took in developing a reference architecture and build. Included is a discussion of major architecture design considerations, explanation of security characteristic achieved by the reference design and a mapping of security characteristics to applicable standards and security control families.

For parties interested in adopting all or part of the NCCoE reference architecture, this guide includes a detailed description of the installation, configuration and integration of all components.
KEYWORDS

access control; access management; attribute provider; authentication; authorization; identity federation; identity management; Identity Provider; relying party

ACKNOWLEDGMENTS

We gratefully acknowledge the contributions of the following individuals and organizations for their generous contributions of expertise, time, and products.

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nate Lesser</td>
<td>NIST National Cybersecurity Center of Excellence</td>
</tr>
<tr>
<td>Paul Timmel</td>
<td>NIST National Cybersecurity Center of Excellence</td>
</tr>
<tr>
<td>Paul Grassi</td>
<td>NIST National Strategy for Trusted Identities in Cyberspace</td>
</tr>
<tr>
<td>Mike Garcia</td>
<td>NIST National Strategy for Trusted Identities in Cyberspace</td>
</tr>
<tr>
<td>Naomi Lefkovitz</td>
<td>NIST National Strategy for Trusted Identities in Cyberspace</td>
</tr>
<tr>
<td>Rene Peralta</td>
<td>NIST National Strategy for Trusted Identities in Cyberspace</td>
</tr>
<tr>
<td>Dave Ferriolo</td>
<td>NIST Computer Security Division</td>
</tr>
<tr>
<td>Vincent Hu</td>
<td>NIST Computer Security Division</td>
</tr>
<tr>
<td>Roger Wiggenstam</td>
<td>NextLabs Inc</td>
</tr>
<tr>
<td>John Conduit</td>
<td>NextLabs Inc</td>
</tr>
<tr>
<td>Srikanth Karanam</td>
<td>NextLabs Inc</td>
</tr>
<tr>
<td>Adam Madlin</td>
<td>Symantec Corporation</td>
</tr>
<tr>
<td>Steve Kruse</td>
<td>Symantec Corporation</td>
</tr>
<tr>
<td>Steve Schmalz</td>
<td>RSA</td>
</tr>
<tr>
<td>Ben Smith</td>
<td>RSA</td>
</tr>
<tr>
<td>Andrew Whelchel</td>
<td>RSA</td>
</tr>
<tr>
<td>Chris Leggett</td>
<td>Ping Identity</td>
</tr>
<tr>
<td>Paul Fox</td>
<td>Microsoft Corporation</td>
</tr>
<tr>
<td>Derek Keatley</td>
<td>Microsoft Corporation</td>
</tr>
<tr>
<td>Hemma Prafullchandra</td>
<td>Hytrust</td>
</tr>
<tr>
<td>John McLeese</td>
<td>Hytrust</td>
</tr>
<tr>
<td>Name</td>
<td>Organization</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Dave Cox</td>
<td>ID/Dataweb</td>
</tr>
<tr>
<td>Chris Donovan</td>
<td>ID/Dataweb</td>
</tr>
</tbody>
</table>
## Contents

1. **Introduction** .................................................................................................................. 1
   1.1 Practice Guide Structure .................................................................................................. 2
   1.2 Typographical Conventions .................................................................................................3

2. **Setting up the Identity Provider** ............................................................................................. 5
   2.1 Introduction ....................................................................................................................... 6
   2.2 Components ....................................................................................................................... 6
      2.2.1 Microsoft AD ............................................................................................................... 6
      2.2.2 PingFederate-IdP ...................................................................................................... 7
      2.2.3 PingFederate-RP ........................................................................................................ 7
      2.2.4 RSA Adaptive Authentication ...................................................................................... 7
      2.2.5 SCE Plugin .................................................................................................................. 8
   2.3 Install Microsoft AD ........................................................................................................... 8
   2.4 Create a User in Microsoft AD ............................................................................................ 9
      2.4.1 Create the LDAP User for Federated Authentication ................................................... 11
   2.5 Install RSA AA .................................................................................................................... 11
   2.6 Configure RSA AA Rules .................................................................................................... 15
      2.6.1 Create Rule for Non-Persistent User Enrollment ............................................................... 15
      2.6.2 Create Rule for Persistent User Enrollment ................................................................. 18
      2.6.3 Create Rule for User Updates ....................................................................................... 18
      2.6.4 Create Rule for Challenge SMS ...................................................................................... 18
      2.6.5 Increase SMS Token Length ......................................................................................... 19
      2.6.6 Create Policy for Session Sign-In .................................................................................. 19
      2.6.7 Create Lists for Session Sign-In .................................................................................... 20
      2.6.8 Create Rules for Session Sign-In ................................................................................... 20
      2.6.9 Create Rule to Allow Forced Sign-In for Payment ......................................................... 22
      2.6.10 Create Custom Fact .................................................................................................... 22
   2.7 Installing and Configuring PingFederate-RP ......................................................................... 23
   2.8 Install PingFederate-IdP ..................................................................................................... 29
   2.9 Install the SCE Plugin for the PingFederate-IdP .................................................................. 29
   2.10 Configure PingFederate-IdP ............................................................................................. 29
      2.10.1 Configure SAML Protocol .......................................................................................... 30
      2.10.2 Create Data Store for Microsoft AD ............................................................................. 31
      2.10.3 Create Credential Validator for Microsoft AD ................................................................. 32
      2.10.4 Create IdP Adapter for Authentication with Microsoft AD via Web Browser Form ........... 35
      2.10.5 Create IdP Adapter for Two-factor Authentication with RSA AA ...................................... 38
      2.10.6 Create Composite IdP Adapter Integrating Microsoft AD and RSA AA ................................ 41
      2.10.7 Configure the Federation Connection to the Relying Party ........................................... 44
   2.11 Certificates ....................................................................................................................... 59
      2.11.1 Certificate Configuration PingFederate ......................................................................... 59
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.12</td>
<td>Functional Test of All Configurations for this Chapter</td>
<td>60</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>66</td>
</tr>
<tr>
<td>3.2</td>
<td>Components</td>
<td>66</td>
</tr>
<tr>
<td>3.2.1</td>
<td>PingFederate-IdP</td>
<td>66</td>
</tr>
<tr>
<td>3.2.2</td>
<td>PingFederate-RP</td>
<td>67</td>
</tr>
<tr>
<td>3.3</td>
<td>Export Metadata from the Identity Provider</td>
<td>67</td>
</tr>
<tr>
<td>3.4</td>
<td>Configure PingFederate-RP Connection to the PingFederate-IdP</td>
<td>70</td>
</tr>
<tr>
<td>3.5</td>
<td>Functional Test of All Configurations for this Chapter</td>
<td>93</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>98</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Components Used in this How-To Guide</td>
<td>98</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Required or Recommended Files, Hardware, and Software</td>
<td>99</td>
</tr>
<tr>
<td>4.2</td>
<td>Installation of required components</td>
<td>100</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Installing SQL Server 2012</td>
<td>100</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Installing IIS 8.0 on the SharePoint Server</td>
<td>100</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Installing Microsoft SharePoint Server 2013</td>
<td>100</td>
</tr>
<tr>
<td>4.3</td>
<td>Creating the Web Application (IIS site) in SharePoint</td>
<td>100</td>
</tr>
<tr>
<td>4.4</td>
<td>Creating and installing SSL certificate</td>
<td>108</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Self-Signed Certificates</td>
<td>108</td>
</tr>
<tr>
<td>4.4.1.1</td>
<td>Creating a Self-Signed Certificate on IIS 8</td>
<td>108</td>
</tr>
<tr>
<td>4.4.1.2</td>
<td>Importing Self-Signed Certificate to SharePoint Certificate Store</td>
<td>110</td>
</tr>
<tr>
<td>4.4.1.3</td>
<td>Add the Self Signed Certificate to Trust management in Central Administration</td>
<td>114</td>
</tr>
<tr>
<td>4.4.1.4</td>
<td>Configure IIS Binding for the Self-Signed Certificate</td>
<td>123</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Certificates Signed by Local or Online Certificate Authority</td>
<td>126</td>
</tr>
<tr>
<td>4.4.2.1</td>
<td>Generating a Certificate Signing Request (CSR)</td>
<td>127</td>
</tr>
<tr>
<td>4.4.2.2</td>
<td>Installing the new signed SSL Certificate</td>
<td>131</td>
</tr>
<tr>
<td>4.4.2.3</td>
<td>Configure the CA-Signed Certificate</td>
<td>133</td>
</tr>
<tr>
<td>4.5</td>
<td>Creating a site collection</td>
<td>133</td>
</tr>
<tr>
<td>4.6</td>
<td>Creating new sub-sites</td>
<td>139</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>144</td>
</tr>
<tr>
<td>5.2</td>
<td>Usage Notes on PingFederate</td>
<td>144</td>
</tr>
<tr>
<td>5.3</td>
<td>Configure a SharePoint Federated Logon Provider</td>
<td>145</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Setting up the Certificate</td>
<td>145</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Configuring the Trusted Identity Token Issuer</td>
<td>149</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Configuring the Token Issuer as a Sign On Option</td>
<td>150</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Configuring the Access Control Rule on SharePoint</td>
<td>151</td>
</tr>
<tr>
<td>5.3.5</td>
<td>Functional Test of the Federated Logon at the Resource Provider</td>
<td>153</td>
</tr>
</tbody>
</table>
5.4 Configure the PingFederate-RP Connection to SharePoint ......................................................157
5.5 Functional Test of All Configurations for This Chapter .............................................................171
5.6 Troubleshooting SharePoint Federated Authentication Problems .............................................175

6 Attribute Exchange Between the Identity Provider and Relying Party ........................................179

6.1 Introduction ...............................................................................................................................180
6.2 Create Custom User Attributes in Microsoft AD ........................................................................180
   6.2.1 Preparing the AD Schema for Creating New Custom Attributes ........................................180
   6.2.1.1 Backing up Your Directory before Making Schema Changes .........................................180
   6.2.1.2 Reviewing Existing Attributes to Avoid Redundancies when Creating New Attributes ....182
   6.2.1.3 Creating New Custom Attributes ....................................................................................184
   6.2.1.4 Generating an ID to Enter into the Unique X500 Object ID Field ....................................185
   6.2.1.5 Adding the New Attribute to the User Class .................................................................186
   6.2.2 Set Values for Custom User Attributes in Microsoft AD ....................................................187
   6.2.2.1 Adding New Columns to the Users Query View ............................................................191
6.3 Configure PingFederate Servers to Pull User Attributes ............................................................193
   6.3.1 Configure PingFederate-IdP to Pull User Attributes During Authentication .......................193
   6.3.1.1 Functional Test of Pulling User Attributes During Authentication .....................................200
   6.3.2 Configure PingFederate-IdP to Pull Environmental Attributes During Authentication ..........202
   6.3.2.1 Configuring a New Data Store that Connects to the RSA Database ..................................205
   6.3.2.2 Modifying the SP Connection to the RP to Add New Environmental Attribute ..............207
   6.3.2.3 Functional Test of Pulling Environmental Attributes During Authentication ....................213
   6.3.3 Configure PingFederate-RP to Pull Attributes from the Identity Provider's SAML Exchange ....214
6.4 Configure PingFederate-RP and SharePoint to Pass and Read Attributes ..................................220
   6.4.1 Configure PingFederate-RP to Pass Attributes to SharePoint .............................................220
   6.4.1.1 Functional Test of PingFederate-RP Passing Attributes to SharePoint ...............................222
   6.4.2 Configure SharePoint to Read Custom Attributes from PingFederate-RP .........................226
   6.4.2.1 Functional Test of SharePoint Reading Attributes from PingFederate-RP ..........................228
6.5 Configure the Claims Viewer Web Part at the SharePoint Site .................................................231
   6.5.1 Configure SharePoint Claims Viewer ..................................................................................233
6.6 Functional Test of All Configurations for this Chapter ..............................................................237
   6.6.1 Temporarily Disable SAML Encryption for Testing and Troubleshooting Message Exchanges ...240
   6.6.1.1 Disable SAML Encryption .............................................................................................240
   6.6.1.2 Enable SAML Encryption Again .....................................................................................241

7 Setting up NextLabs to Protect SharePoint .................................................................................245

7.1 Introduction ...............................................................................................................................246
7.2 Components ............................................................................................................................246
   7.2.1 NextLabs Control Center (release 7.5) .................................................................................246
   7.2.2 NextLabs Policy Studio: Enterprise Edition .........................................................................247
   7.2.3 NextLabs Policy Controller ...............................................................................................247
   7.2.4 NextLabs Entitlement Manager for Microsoft SharePoint Server ......................................249
   7.2.5 Required or Recommended Files, Hardware, and Software .............................................249
7.3 Installation and Configuration of NextLabs Control Center (on the SQL Server) .......................252
8 Defining Policies and Enforcing Access Decisions with NextLabs................................. 287

8.1 Introduction .................................................................................................................. 288
8.1.1 Components and Sub-components Used in this How-To Guide ................................ 288
8.1.2 Pre-requisites to Complete Prior to This How-To Guide ........................................ 288

8.2 Policy Strategy .............................................................................................................. 289
8.2.1 Top-level Blacklisting Deny Policy, Whitelisting Allow Sub-policies ....................... 289
8.2.2 Global Policies ........................................................................................................ 289

8.3 Translation of Business Logic into Policy..................................................................... 290
8.3.1 ABAC Build Scenario - Runabout Air Business Rules ............................................. 290
8.3.2 Translation of Runabout Air Business Rules into ABAC Policies ............................. 290

8.4 Using the NextLabs Policy Studio GUI for Policy Definition and Deployment .......... 291
8.4.1 Login and Initial Screen in Policy Studio ................................................................. 292
8.4.2 Policy Studio Menu Commands .............................................................................. 294
8.4.3 Defining and Deploying Components ..................................................................... 294
8.4.3.1 Explanation of Components in NextLabs ................................................................. 294
8.4.3.2 Defining and Deploying User Components ............................................................... 295
8.4.3.3 Defining and Deploying Resource Components ...................................................... 302
8.4.4 Defining Policy ........................................................................................................ 306
8.4.4.1 Creating a Policy Set Folder .................................................................................... 306
8.4.4.2 Defining Department-based Policy Set .................................................................. 308
8.4.4.3 Defining a Sensitive-based Policy Set ..................................................................... 312
8.4.4.4 Defining a Maintenance-based Policy Set .............................................................. 320
8.4.4.5 Defining an IP Address-based Policy Set ............................................................... 324
8.4.5 Deploying Policy ..................................................................................................... 329

7.4.1 Installation ................................................................................................................ 267

7.5 Installation and Configuration of Policy Controller (PDP) ............................................ 273
7.5.1 Installation ................................................................................................................ 273

7.6 Installation and Configuration of NextLabs Entitlement Manager for SharePoint Server ....................................................................................................................... 276
7.6.1 Installation and Configuration .................................................................................... 276
7.6.1.1 Verify that a Web Application Site and Site Collection Already Exist in SharePoint ... 276
7.6.1.2 Install NextLabs Entitlement Manager for SharePoint Server ................................. 277
7.6.1.3 Deploy Entitlement Manager for SharePoint Server to your SharePoint Farm ......... 279
7.6.1.4 Enable Policy Enforcement on your Web Application via SharePoint Central Administration 282

7.7 Functional Tests ......................................................................................................... 284
7.7.1 Verify that the NextLabs Webpart for Policy Enforcement has Successfully Been Enabled on the Site Collection in SharePoint 284
7.7.2 Test to Verify the NextLabs Service is Running ....................................................... 286

7.3.1 Installation and Configuration .................................................................................... 252
7.3.1.1 Install the Microsoft SQL Server via Microsoft SQLServer 2012 .............................. 252
7.3.1.2 Create a New Database and Database User for the NextLabs Control Center Installation and Administration 252
7.3.1.3 Install and Configure the NextLabs Control Center .................................................. 256

8.4.6 Modifying and Re-Deploying Policies and Components .......................................................... 331
8.4.6.1 Modifying and Deploying Existing Policies ........................................................................ 331
8.4.6.2 Modifying and Deploying Existing Components ................................................................. 331
8.4.7 Deactivating Policies and Components ................................................................................... 332
8.4.7.1 Deactivating Policies ........................................................................................................ 332
8.4.7.2 Deactivating Components .................................................................................................. 333
8.4.8 Deleting Policies and Components ........................................................................................ 333
8.4.8.1 Deleting Policies ............................................................................................................. 334
8.4.8.2 Deleting Components ...................................................................................................... 334
8.5 Configuring Attributes in NextLabs ............................................................................................ 334
8.5.1 Stopping the NextLabs Policy Controller Service ................................................................. 334
8.5.2 Editing the Configuration File ................................................................................................ 335
8.5.2.1 Locating and Opening the SharePoint Enforcer configuration.xml File ........................................ 335
8.5.2.2 Configuring Resource Attributes from SharePoint Metadata .............................................. 336
8.5.2.3 Configuring User Attributes from SharePoint Claims ............................................................ 336
8.5.2.4 Saving Changes to the Configuration File ........................................................................... 338
8.5.3 Restarting IIS via Windows PowerShell ..................................................................................... 338
8.5.4 Restarting the NextLabs Policy Controller Service ................................................................. 339
8.6 Functional Test .......................................................................................................................... 339
8.6.1 Updated bin file after Policy Creation/modification ................................................................. 339
8.6.1.1 On the SharePoint Server note timestamp of the bundle.bin file and decrypt its contents .... 339
8.6.2 Reviewing NextLabs AgentLog to Illustrate History of Access Control Evaluations During SharePoint 342

9 Leveraging NextLabs Control Center Reporter for Reporting and Auditing Purposes ... 347

9.1 Introduction ................................................................................................................................ 348
9.1.1 Components Used in this How-To Guide ............................................................................... 348
9.1.2 Pre-requisites to Complete Prior to This How-To Guide ......................................................... 348
9.2 Introduction to NextLabs Control Center Reporter ................................................................. 349
9.2.1 Opening Reporter ................................................................................................................. 349
9.3 Introduction to Reporter Dashboard ........................................................................................... 350
9.3.1 Exploring the Dashboard ...................................................................................................... 352
9.4 Introduction to Defining and Running Custom Reports in Reporter .......................................... 354
9.4.1 Defining a Custom Report ....................................................................................................... 355
9.4.1.1 Logging into Reporter ......................................................................................................... 356
9.4.1.2 Defining the Custom Report ............................................................................................. 357
9.4.1.3 Setting the Custom Report Display Options ...................................................................... 360
9.4.2 Running a Custom Report ...................................................................................................... 361
9.5 Example Custom Report and Available Formats ................................................................. 361
9.5.1 Defining the Example Custom Report ..................................................................................... 361
9.5.1.1 Customizing Report Query Fields for this Report .............................................................. 361
9.5.1.2 Editing the Columns for Custom Views ............................................................................ 362
9.5.1.3 Running the Report Query ................................................................................................. 363
10 Configuring a Secondary Attribute Provider ................................................................. 375

10.1 Introduction ............................................................................................................... 376
  10.1.1 Prerequisites ....................................................................................................... 377
  10.1.2 Criteria for Secondary Attribute Collection ...................................................... 377
  10.1.3 Components ....................................................................................................... 377
    10.1.3.1 Sequence Diagram of Custom PIP Component Interactions ............................ 379

10.2 Component Software and Hardware Requirements .................................................. 381

10.3 Ping Custom Data Store ............................................................................................. 382
  10.3.1 Functionality and Architecture .......................................................................... 382
  10.3.2 Deploying the Ping Custom Data Store .............................................................. 383
  10.3.3 Compilation ........................................................................................................ 384
  10.3.4 Configuration within PingFederate Administrative Console ............................... 385

10.4 NextLabs PIP Plugin ................................................................................................. 388
  10.4.1 Architecture ...................................................................................................... 388
    10.4.1.1 Required classes of the NextLabs PIP Plugin: ................................................... 388
    10.4.1.2 Other Required Files or Deployment Notes: ....................................................... 389
    10.4.1.3 Notes on Jar and Properties File Deployment within NextLabs Policy Controller Software Architecture: 390
  10.4.2 Understanding how the NextLabs PIP Plugin interacts with Build Components ............ 390
  10.4.3 Compilation and Deployment ............................................................................... 391
    10.4.3.1 Compiling the NextLabs PIP Plugin Jar ............................................................ 391
    10.4.3.2 Stopping the NextLabs Policy Controller Service Before NextLabs PIP Plugin Jar Deployment: 391
    10.4.3.3 Deploying the NextLabs PIP Plugin Jar and its Configuration File .................... 392
    10.4.3.4 Resetting IIS and Restarting the NextLabs Policy Controller Service ................ 393

10.5 Protocol Broker ....................................................................................................... 393
  10.5.1 Architecture ...................................................................................................... 393
    10.5.1.1 Communication Between NextLabs PIP Plugin and Protocol Broker ................. 394
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.5.1.2 Communication Between Protocol Broker and PingFederate RP Server</td>
<td>394</td>
</tr>
<tr>
<td>10.5.2 Deployment</td>
<td>396</td>
</tr>
<tr>
<td>10.5.2.1 System and Environment Requirements</td>
<td>396</td>
</tr>
<tr>
<td>10.5.2.2 Configuration</td>
<td>396</td>
</tr>
<tr>
<td>10.5.2.3 Preparation and Compilation</td>
<td>397</td>
</tr>
<tr>
<td>10.5.3 Example SAML Request and Response Output</td>
<td>400</td>
</tr>
<tr>
<td>10.5.3.1 Example of Tomcat Output from our Build that Illustrates a SAML Request</td>
<td>400</td>
</tr>
<tr>
<td>10.5.3.2 Example of Tomcat Output from our Build that Illustrates a SAML Response</td>
<td>401</td>
</tr>
<tr>
<td>10.6 Apache Directory Service (ApacheDS)</td>
<td>402</td>
</tr>
<tr>
<td>10.6.1 Layout</td>
<td>403</td>
</tr>
<tr>
<td>10.6.2 Download</td>
<td>403</td>
</tr>
<tr>
<td>10.6.2.1 Verify the Integrity of the Downloaded File</td>
<td>404</td>
</tr>
<tr>
<td>10.6.3 Installation</td>
<td>404</td>
</tr>
<tr>
<td>10.6.3.1 Functional Test of the ApacheDS Installation</td>
<td>408</td>
</tr>
<tr>
<td>10.6.4 Start and Stopping the Server</td>
<td>409</td>
</tr>
<tr>
<td>10.6.5 ApacheDS Configuration</td>
<td>409</td>
</tr>
<tr>
<td>10.7 PingFederate - Apache Integration</td>
<td>409</td>
</tr>
<tr>
<td>10.7.1 Provisioning of Server Credential</td>
<td>410</td>
</tr>
<tr>
<td>10.7.1.1 Creation of Server Connection</td>
<td>410</td>
</tr>
<tr>
<td>10.7.1.2 Verification</td>
<td>419</td>
</tr>
<tr>
<td>10.7.1.3 Configuration Steps on PingFederate RP Server</td>
<td>421</td>
</tr>
<tr>
<td>10.7.1.4 Creation of Data Store to Connect to ApacheDS</td>
<td>423</td>
</tr>
<tr>
<td>10.8 Configuration of PingFederate to Query the JIT Cache when Responding to Secondary Attribute Requests</td>
<td>425</td>
</tr>
<tr>
<td>10.8.1 Introduction</td>
<td>425</td>
</tr>
<tr>
<td>10.8.2 Prerequisites</td>
<td>426</td>
</tr>
<tr>
<td>10.8.2.1 SP Connection</td>
<td>426</td>
</tr>
<tr>
<td>10.8.2.2 IDP Connection</td>
<td>441</td>
</tr>
<tr>
<td>10.9 ApacheDS Schema Extension</td>
<td>445</td>
</tr>
<tr>
<td>10.9.1 Pre-Requisites</td>
<td>445</td>
</tr>
<tr>
<td>10.9.2 Procedure</td>
<td>445</td>
</tr>
<tr>
<td>10.10 Functional Tests</td>
<td>459</td>
</tr>
<tr>
<td>10.10.1 Testing the Ready State of the NextLabs Policy Controller Service</td>
<td>460</td>
</tr>
<tr>
<td>10.10.2 Test the Successful Loading of the Custom Plugin within the NextLabs Policy Controller Software Architecture</td>
<td>460</td>
</tr>
<tr>
<td>10.10.3 Testing that the Protocol Broker .war File Loads Correctly in Tomcat Server</td>
<td>462</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Practice Guide Structure ................................................................. 2
1.2 Typographical Conventions ......................................................... 3
1.1 Practice Guide Structure

This NIST Cybersecurity Practice Guide demonstrates a standards-based example solution and provides users with the information they need to replicate this approach to implementing attribute based access control (ABAC) that leverages identity federation. The example solution is modular and can be deployed in whole or in parts.

This guide contains three volumes:

- **NIST SP 1800-3a: Executive Summary**
- **NIST SP 1800-3b: Approach, Architecture, and Security Characteristics** - what we built and why
- **NIST SP 1800-3c: How To Guides** - instructions for building the example solution - this document

The following instructions show IT professionals and security engineers how the National Cybersecurity Center of Excellence (NCCoE) implemented an example solution to the challenge of implementing an ABAC deployment that supports identity federation. We developed a build that conforms to federal standards and best practices, and addresses the challenge of providing access control mechanisms for a diverse set of subjects requesting access to corporate resources when many of these subjects may not be managed or even known to the enterprise. This build also helps ensure that once users are authenticated, fine-grained access decisions are enforced based on a range of attributes, such as user identity, resource type, and environmental conditions.

This example solution is packaged as a “How To” guide. The guide demonstrates how to implement standards-based, commercially available cybersecurity technologies in the real world, based on risk analysis. We cover all the products that we employed in this example solution. We do not recreate the product manufacturers' documentation, which is generally widely available. Rather, we show how we incorporated the products together in our environment to create the example solution.

This guide assumes that the IT professionals using this document have experience implementing security products within an enterprise. While we have used a suite of commercial products to address this challenge, this guide does not endorse these particular products. We assume that you have the knowledge and expertise to choose other products that might better fit your IT systems and business processes. If you use substitute products, we hope you'll seek products that are congruent with standards and best practices, as we have. Refer to **NIST SP 1800-3b: Approach, Architecture, and Security Characteristics**, Section 4.5, table 4.2 for a list of the products that we used, mapped to the cybersecurity controls provided by this example solution, to understand the characteristics you should seek in alternate products. Section 4.4, Security Characteristics and Controls Mapping, of that document describes how we arrived at this list of controls.

This NIST Cybersecurity Practice Guide does not describe “the” solution, but a possible solution. This is a draft version. We are seeking feedback on its contents and welcome your

---

1. Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST or NCCoE, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.
input. Comments and suggestions will improve subsequent versions of this guide. Please contribute your thoughts to abac-nccoe@nist.gov, and join the discussion at http://nccoe.nist.gov/forums/attribute-based-access-control.

1.2 Typographical Conventions

The following table presents typographic conventions used in this volume.

<table>
<thead>
<tr>
<th>Typeface/ Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Italics</em></td>
<td>references to documents that are not hyperlinks, new terms, and placeholders</td>
<td>For detailed definitions of terms, see the NCCoE Glossary.</td>
</tr>
<tr>
<td><strong>Bold</strong></td>
<td>names of menus, options, command buttons and fields</td>
<td>Choose File &gt; Edit.</td>
</tr>
<tr>
<td>Monospace</td>
<td>command-line input, on-screen computer output, sample code examples, status codes</td>
<td>mkdir</td>
</tr>
<tr>
<td><strong>Monospace Bold</strong></td>
<td>command-line user input contrasted with computer output</td>
<td>service sshd start</td>
</tr>
<tr>
<td>blue text</td>
<td>link to other parts of the document, a web URL, or an email address</td>
<td>All publications from NIST's National Cybersecurity Center of Excellence are available at <a href="http://nccoe.nist.gov">http://nccoe.nist.gov</a></td>
</tr>
</tbody>
</table>
## Setting up the Identity Provider

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Introduction</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Components</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Install Microsoft AD</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Create a User in Microsoft AD</td>
<td>9</td>
</tr>
<tr>
<td>2.5</td>
<td>Install RSA AA</td>
<td>11</td>
</tr>
<tr>
<td>2.6</td>
<td>Configure RSA AA Rules</td>
<td>15</td>
</tr>
<tr>
<td>2.7</td>
<td>Installing and Configuring PingFederate-RP</td>
<td>23</td>
</tr>
<tr>
<td>2.8</td>
<td>Install PingFederate-IdP</td>
<td>29</td>
</tr>
<tr>
<td>2.9</td>
<td>Install the SCE Plugin for the PingFederate-IdP</td>
<td>29</td>
</tr>
<tr>
<td>2.10</td>
<td>Configure PingFederate-IdP</td>
<td>29</td>
</tr>
<tr>
<td>2.11</td>
<td>Certificates</td>
<td>59</td>
</tr>
<tr>
<td>2.12</td>
<td>Functional Test of All Configurations for this Chapter</td>
<td>60</td>
</tr>
</tbody>
</table>
2.1 Introduction

This guide details an attribute based access control (ABAC) implementation that leverages identity federation. In a federation model, the Identity Provider authenticates the user requesting access and provides attributes assigned to that user to the Relying Party. The Relying Party, which controls access to the resource requested by the user, utilizes the identity and attributes information to make run-time decisions to grant or deny access to the user.

In this chapter we install and configure federation components at the Identity Provider. The components described in this chapter facilitate federated, SAML-based authentication using account credentials in the Identity Provider’s Microsoft Active Directory Domain Services (referred to as Microsoft AD in this guide). The federated authentication between the Relying Party and the Identity Provider is facilitated by Ping Identity’s PingFederate application. This build also requires the user to authenticate with a second factor, which is handled by the RSA adaptive authentication server.

Each of the components used for the build are described in section 2.2, Components. Following that section are step-by-step instructions for installing, configuring, and integrating the components. If you follow the instructions in this chapter, you will be able to perform a functional test to verify the successful completion of the steps for installing, configuring, and integrating the components.

2.2 Components

Federated Authentication at the Identity Provider involves the following distinct components:

- **Microsoft AD**: An LDAP directory service that stores user account and attribute information.
- **PingFederate-IdP**: A federation system or trust broker for the Identity Provider.
- **PingFederate-RP**: Serves as the trust broker for SharePoint.
- **RSA Adaptive Authentication (RSA AA)**: Requires the user to authentication using an SMS message sent to their mobile phone. Collects environmental information about the user and the user’s system or agent at the time of authentication.
- **SCE Plugin**: Handles communications between the PingFederate-IdP and the RSA AA.

2.2.1 Microsoft AD

Microsoft AD acts as a user identity management repository for the IdP. This includes the ability to provision and de-provision user identities; the creation, modification, and deletion of subject attributes; and the provisioning and de-provisioning of subject attributes to specific user identities. In this build, Microsoft AD is the only source for subject attributes.
2.2.2 PingFederate-IdP

Ping Identity PingFederate-IdP serves as a federation system or trust broker for the IdP. PingFederate-IdP provides initial user authentication and retrieval of user attributes to satisfy SAML requests from the RP. Once the user has been authenticated, PingFederate-IdP queries subject attributes from AD and environmental attributes from the RSA AA event log. PingFederate-IdP packages both subject and environmental attributes in a SAML 2.0 token to be sent to the RP.

PingFederate Usage Notes

- When using the PingFederate application to perform an administrative configuration, there is usually a sequence of screens that require user entry, ending with a summary page. Once you click Done on the summary page, you must also click Save on the following page to actually save the configurations. If you forget to click Save, you may inadvertently lose changes to the configuration.

- In the PingFederate application and associated documentation, the Relying Party is referred to as the Service Provider.

- When using the PingFederate application to perform configuration, refer to the title of the tab with a small star icon to its left, to identify the item you are currently configuring. For example, if you navigated to the following screen, you would be on the IdP Adapter screen.

2.2.3 PingFederate-RP

Ping Identity PingFederate-RP serves as the trust broker for SharePoint. When the user requires authentication, PingFederate-RP redirects the user to the IdP via a SAML request to get the necessary assertions. Once authenticated, PingFederate-RP arranges for the browser’s HTTPS content to have the proper information in proper format for acceptance at the target resource (SharePoint).

2.2.4 RSA Adaptive Authentication

RSA Adaptive Authentication (RSA AA) has the responsibility to gather environmental information about the user and the user’s system or agent at the time of authentication. RSA AA collects information such as patch level, operating system, and location, and it generates a risk score associated with the user authentication. A risk score threshold can then be defined in RSA AA, which, if exceeded, can force a user to step up to one of the additional authentication mechanisms. In this build, information collected by RSA AA to generate a risk score is also passed through PingFederate-IdP to the RP side of the operation to be used as environmental attributes. The RSA AA event log contains the transaction ID of each user authentication and the associated environmental information collected by RSA AA at the time of authentication.
2.2.5 SCE Plugin

The SCE Plugin handles communications between the PingFederate-IdP and the RSA AA. It is responsible for passing the RSA AA transaction ID for the user authentication that PingFederate-IdP uses to query the RSA AA event log.

Table 2.1 Required or Recommended Files, Hardware, and Software

<table>
<thead>
<tr>
<th>Component</th>
<th>Required Files</th>
<th>Recommanded or Minimum Hardware Requirements</th>
<th>Hardware Used in this Build</th>
<th>Recommended or Minimum Operating System or Other Software</th>
<th>Operating System or Other Software Used in this Build</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft AD</td>
<td></td>
<td>512MB RAM; 1.4GHz CPU; 32GB free disk space</td>
<td>4GB RAM; 2.2GHz CPU; 108GB free disk space</td>
<td></td>
<td>Microsoft Windows Server 2012</td>
</tr>
<tr>
<td>PingFederate SCE</td>
<td>sce-adapters-pingfederate-aa.1.1.jar</td>
<td>1GB RAM; 1.8GHz CPU; 250MB free disk space</td>
<td>4GB RAM; 2.2GHz CPU; 98 GB</td>
<td>sce-adapters-pingfederate-aa.1.1.jar</td>
<td>Microsoft Windows Server 2012</td>
</tr>
<tr>
<td>RSA AA</td>
<td>Adaptive Authentication (On-Premise) 7.0.0.0-SNAPSHOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.3 Install Microsoft AD

1. Log on to the server that will host Microsoft AD.

2. Follow the instructions at the link below to create a new Microsoft AD domain that will store the accounts and identity information for the Identity Provider.

3. During setup, you will be asked to provide a name for your new domain. The name of the domain used for this build is ABAC.TEST.

2.4 Create a User in Microsoft AD

To create a user account in the Microsoft AD Domain:

1. Launch the Active Directory Users and Computers program.

2. Click on the name of your domain in the left pane and then right-click on the Users folder in the right pane.

3. In the popup menu that appears, select New, and then select User.
4. In the New Object - User screen that displays, type the First and Last name of the user, as well as their User logon name (that is, the account name).

5. Click Next.

6. In the password screen that appears, type in the user’s initial password. Then, type it again in the Confirm password field. When users log in for the first time, they will be prompted to create her own unique password.

7. Click Next.

8. In the confirmation screen with information about the new user that displays, click Finish to complete the operation.

   When the user logs on to the domain for the first time, the user will be prompted to create a new unique password.

   The following illustrations demonstrate what the new password screens may look like on Microsoft Windows Server 2012 when the user Lucy Smith attempts to log on to a computer in the ABAC.TEST domain using her user name Ismith and the initial password.
Chapter 2. Setting up the Identity Provider

When Lucy clicks OK, she will see the screen below. She will type in her new password, which adheres to the organization’s password strength policy, then she will type the password in again to confirm.

When she presses Enter, Microsoft Windows will change her password.

2.4.1 Create the LDAP User for Federated Authentication

Follow the steps in the previous section to create a user named LDAP user in Microsoft AD. This user account will be used by the PingFederate-IdP to perform LDAP queries in Microsoft AD.

2.5 Install RSA AA

RSA AA (On-Premise) comes packaged as a virtual snapshot that will must be installed on a virtual machine. A full installation requires core and back office applications, database scripts and maintenance tools - all necessary for this build. Follow these instructions to install RSA AA for the Identity Provider.

1. Log on to VMware and load the RSA AA virtual appliance. [e.g. Adaptive Authentication (On-Premise) 7.0.0.0-SNAPSHOT]
2. Start the RSA AA virtual machine using VMware.
3. Log on to the server that hosts the new virtual machine.

4. Launch the RSA AA installation file.

5. On the Installation Types screen, select **Full** to install all required components. Then, click **Next**.

6. Click **Next** in the Installation Components screen.
7. In the environment screen, set the database type [MS SQL] and the JDBC driver file. This is illustrated in the following figure.

8. For the core database setup, create a new database, and set the core database properties and credentials.
9. On the Core Database screen, set parameters for the data and log files (directory, name, size, and growth).

![Core Database screen image]

10. On the Core Applications screen, provide the web service credentials and application server properties.

![Core Applications screen image]

11. Review the configuration options on the Installation Parameters Summary and click **Install**. Once complete, you can confirm that the installation was successful by viewing the log files.
2.6 Configure RSA AA Rules

RSA has a built-in policy management application that allows administrators to create and update rules for user login based on various scenarios. For example, high-risk users can be required to answer challenge questions or respond to an out-of-band SMS. For more information, see the Back Office User’s Guide. This example shows how to create a challenge rule for users to confirm identity for large transactions using an out-of-band SMS code. RSA Back Office allows administrators to manage set up policy for enabling the enhanced features provided by the RSA adapter such as answering challenge questions and providing SMS confirmation codes are enabled through this interface.

2.6.1 Create Rule for Non-Persistent User Enrollment

RSA AA requires information for each user to help verify their identity. These users are classified into two groups: persistent and non-persistent users. A rule is created to request enrollment information for non-persistent users, those not kept in the user database.

1. Login to the Back Office application
   [http://xxx.xxx.xxx.xxx:8080/backoffice]
3. In the Rule Details (in the General tab):
   a. Set Rule Name to User Enrollment Not Persistent - Adapter.
   b. Set the Status to Production.
Note: The rule cannot be in production until it is created and approved by an administrator.

c. In Event Type, select Create User and Enroll.

d. Set the Order to 1.

4. Click Next.

5. In the Rule Conditions page add a condition (Condition 1) and with one expression (Expression 1). Set Expression 1 to Account Details such that Persistent User is Equal to FALSE.
6. Click **Next**.

7. In the **Rule Actions** page:
   a. Set **Action** to **Challenge**.
   b. Set **Authentication Methods** to **QUESTION**, **OOBSMS**, **OOBPHONE**, **SECURID**, and **TeleSign2FASms**.
   c. In **Create Case**, make sure that only for when authentication fails is selected. Then, click **Next**.

8. Review the rule settings in the **Summary** page. Then, click **Save and Finish**.
   Once created, a rule is in Work in Progress status until approved by an administrator.

9. Click **Status** and **Approve Status**, then click **Approve** to set rule to **Production** status.

You can use these steps to create each of the rules in the following sections.
2.6.2 **Create Rule for Persistent User Enrollment**

Persistent users are those that will be added to the user table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>User Enrollment Persistent - Adapter</td>
</tr>
<tr>
<td>Event Type</td>
<td>Create User, Enroll</td>
</tr>
<tr>
<td>Rule Order</td>
<td>2</td>
</tr>
<tr>
<td>Rule Condition</td>
<td>IF (Account Details -&gt; Persistent User Equal to TRUE)</td>
</tr>
<tr>
<td>Rule Action</td>
<td>Allow</td>
</tr>
<tr>
<td>Authentication Method</td>
<td></td>
</tr>
<tr>
<td>Create Case</td>
<td>No</td>
</tr>
</tbody>
</table>

2.6.3 **Create Rule for User Updates**

Once users are created, a rule is applied to allow persistent users to update their information.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>User Update</td>
</tr>
<tr>
<td>Event Type</td>
<td>User Update</td>
</tr>
<tr>
<td>Rule Order</td>
<td>3</td>
</tr>
<tr>
<td>Rule Condition</td>
<td>IF (Account Details -&gt; Persistent User Equal to TRUE)</td>
</tr>
<tr>
<td>Rule Action</td>
<td>Allow</td>
</tr>
<tr>
<td>Authentication Method</td>
<td></td>
</tr>
<tr>
<td>Create Case</td>
<td>No</td>
</tr>
</tbody>
</table>

2.6.4 **Create Rule for Challenge SMS**

In this build, large transactions require users to respond to an out-of-band SMS challenge during authentication. When transactions meet the prerequisite, a random code will be sent to the user’ SMS-enabled device that must be entered to confirm the transaction.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>Challenge SMS for Payment</td>
</tr>
<tr>
<td>Event Type</td>
<td>Challenge</td>
</tr>
<tr>
<td>Rule Order</td>
<td>4</td>
</tr>
<tr>
<td>Rule Condition</td>
<td>IF (Transaction Details -&gt; Transaction Amount is BETWEEN 5000 and 10000)</td>
</tr>
</tbody>
</table>
2.6.5 Increase SMS Token Length

The default token length for out-of-band SMS is currently set to four [4] digits. Access the Administration tab on the Back office application. Under Components, select Authentication Methods and scroll down to Out-of-Band SMS section. Adjust the token length by changing the value of SMS - OTP Token Length to six [6].

![Figure 2.1 Out-of-Band Token Length](image)

2.6.6 Create Policy for Session Sign-In

The following rules create different sign-in scenarios for users based on an RSA-generated risk score at the time of login. RSA AA uses a risk engine to give users a risk score to determine a level of trust at the time of access. See the tables below for the session sign-in parameters for each risk level. Before creating the session sign-in rules, lists need to be created to group users together. This build will group users into four categories based on risk level (low, medium, high, and critical).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Action</td>
<td>Allow</td>
</tr>
<tr>
<td>Authentication Method</td>
<td>1. OOBSMS</td>
</tr>
<tr>
<td>Create Case</td>
<td>When Authentication Succeeds</td>
</tr>
</tbody>
</table>

![Table](image)
2.6.7 Create Lists for Session Sign-In

1. Log in to the Back Office application.
2. Go to Policy Management and select Manage Lists.
3. Set List Name to Low Risk Users, List Type to User ID, and Status to Enabled.
4. Under List Content, select Add Value and set the Value to demolowrisk and Organization to default.
5. Click Add Value.
6. Click Save.

Repeat these steps to create a list for Medium, High, and Critical risk users.

![Figure 2.2 List for Session Sign-In Created Successfully](image)

2.6.8 Create Rules for Session Sign-In

Repeat the steps as in section 2.6.1, Create Rule for Non-Persistent User Enrollment, to create the session sign-in rules for different user groups.

### Table 2.2 Session Sign-In - Low Risk

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>Session Sign In - Low Risk</td>
</tr>
<tr>
<td>Event Type</td>
<td>Session Sign-in</td>
</tr>
<tr>
<td>Rule Order</td>
<td>5</td>
</tr>
<tr>
<td>Rule Condition</td>
<td>IF (Account Details-&gt;User ID within Low Risk Users)</td>
</tr>
<tr>
<td>Rule Action</td>
<td>Allow</td>
</tr>
</tbody>
</table>
### Table 2.2 Session Sign-In - Low Risk

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Method</td>
<td></td>
</tr>
<tr>
<td>Create Case</td>
<td>No</td>
</tr>
</tbody>
</table>

### Table 2.3 Session Sign-In - Medium Risk

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>Session Sign In - Medium Risk</td>
</tr>
<tr>
<td>Event Type</td>
<td>Session Sign-in</td>
</tr>
<tr>
<td>Rule Order</td>
<td>6</td>
</tr>
<tr>
<td>Rule Condition</td>
<td>IF (Account Details-&gt;User ID within Medium Risk Users)</td>
</tr>
<tr>
<td>Rule Action</td>
<td>Allow</td>
</tr>
<tr>
<td>Authentication Method</td>
<td>1. Question</td>
</tr>
<tr>
<td>Create Case</td>
<td>When Authentication Fails</td>
</tr>
</tbody>
</table>

### Table 2.4 Session Sign-In - High Risk

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>Session Sign In - High Risk</td>
</tr>
<tr>
<td>Event Type</td>
<td>Session Sign-in</td>
</tr>
<tr>
<td>Rule Order</td>
<td>5</td>
</tr>
<tr>
<td>Rule Condition</td>
<td>IF (Account Details-&gt;User ID within High Risk Users)</td>
</tr>
<tr>
<td>Rule Action</td>
<td>Challenge</td>
</tr>
<tr>
<td>Authentication Method</td>
<td>1. OOBSMS</td>
</tr>
<tr>
<td></td>
<td>2. OOBPhone</td>
</tr>
<tr>
<td>Create Case</td>
<td>When Authentication Fails</td>
</tr>
</tbody>
</table>

### Table 2.5 Session Sign-In - Critical Risk

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>Session Sign In - Low Risk</td>
</tr>
<tr>
<td>Event Type</td>
<td>Session Sign-in</td>
</tr>
<tr>
<td>Rule Order</td>
<td>8</td>
</tr>
<tr>
<td>Rule Condition</td>
<td>IF (Account Details-&gt;User ID within Critical Risk Users)</td>
</tr>
</tbody>
</table>
2.6.9 Create Rule to Allow Forced Sign-In for Payment

The rules for session sign-in in the preceding sections were based predefined facts built within RSA AA. This build requires a rule that uses additional facts that are not within the build. Fortunately, new facts can be created within the Back Office application. Once custom facts are created, they can be used to further build rules.

2.6.10 Create Custom Fact

1. Login in to the Back Office application.
2. Go to Policy Management and select Manage Custom Facts.
3. Select New and set the Field Name to Force Workflow, Field Type to String, and Status to Enabled.
4. Click Save.
5. Create a new rule using this custom fact that allows payment if this fact is met. Use the settings in the following table.

### Table 2.6 Force Allow

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>Force Allow</td>
</tr>
<tr>
<td>Event Type</td>
<td>Payment, Session Sign-in</td>
</tr>
<tr>
<td>Rule Order</td>
<td>9</td>
</tr>
<tr>
<td>Rule Condition</td>
<td>IF (Custom Fact -&gt; Force Workflow Equal to Allow)</td>
</tr>
<tr>
<td>Rule Action</td>
<td>Allow</td>
</tr>
<tr>
<td>Authentication Method</td>
<td></td>
</tr>
<tr>
<td>Create Case</td>
<td>No</td>
</tr>
</tbody>
</table>

### 2.7 Installing and Configuring PingFederate-RP

The PingFederate installation in this section is for the Federation Server at the Relying Party. This is the only component at the Relying Party in this chapter. Even though the goal of this chapter is to setup the federation for the Identity Provider, the basic configuration of the PingFederate-RP in this section is necessary, in order to produce metadata that is exchanged with the Identity Provider. A complete configuration of the PingFederate-RP will be performed in chapter 3 of this guide.

1. Log on to the Relying Party’s server that will host the PingFederate service and follow the instructions at the link below to install PingFederate and run it as a Windows service.
   
   https://documentation.pingidentity.com/display/PF73/Installation

2. Follow the steps in this section to perform a basic configuration of the PingFederate-RP and export the metadata.

3. Launch your browser and navigate to the PingFederate app URL:
   
   https://<DNS_NAME>:9999/pingfederate/app. Replace DNS_NAME with the fully qualified name of the Relying Party’s PingFederate server (e.g. https://rp.abac.test:9999/pingfederate/app).

4. Log on to the PingFederate application using the credentials you configured in the previous installation section.
5. On the Main menu under System Settings, click Server Settings.
6. Click the Roles and Protocols tab.
7. Select Enable Identity Provider (IdP) role and support the following.
8. Select SAML 2.0.
10. Select Enable Service Provider (SP) role and support the following.
11. Select the SAML 2.0.
12. Click Next.
13. On the Federation Info screen, enter the Base URL and SAML 2.0 Entity ID using the format https://<DNS_NAME>:9031 (e.g. https://rp.abac.test:9031).
14. Enter the WS-Federation Realm using the format urn:<DNS_NAME> (e.g. urn:rp.abac.test).
   
   Note: Keep a copy of the urn because it will be used later to configure the WS-Federation relationship with Sharepoint.
15. Click **Save**.

16. On the **Main** menu under **Administrative Functions**, click **Metadata Export**.

17. On the **Metadata Role** screen, select **I am the Service Provider (SP)**.

18. Click **Next**.

19. On the **Metadata Mode** screen, select **Select information to include in metadata manually**.
20. Click **Next**.

21. On the Protocol screen, make sure that **SAML 2.0** is listed.

22. Click **Next**.

23. On the Attribute Contract screen, click **Next**.

24. On the Signing Key screen, select the certificate that will be used to sign communications with the Identity Provider.
25. Click **Next**.

26. On the Metadata Signing screen, if you plan to sign the metadata file that will be exported, select the certificate that will be used to sign the file.

27. Click **Next**.

28. On the XML Encryption Certificate screen, select the certificate that the Identity Provider will use to encrypt XML messages.
29. Click **Next**.

30. Click **Export**.

This will create an export file that contains the metadata of the Relying Party that you can download using the browser. This file will be used later in the chapter, when configuring the PingFederate-IDP.
2.8 Install PingFederate-IdP

This PingFederate installation in this section is for the PingFederate-IdP.

Log on to the server that will host the PingFederate service for the Identity Provider and follow the instructions at the link below to install PingFederate and run it as a Windows service.

https://documentation.pingidentity.com/display/PF73/Installation

2.9 Install the SCE Plugin for the PingFederate-IdP

The SCE Plugin integrates the features provided by RSA AA with PingFederate-IdP by providing a customizable user interface when RSA AA is accessed. New users will be enrolled into RSA’s enhanced security features and be prompted to provide information such as security questions, a phone number, email address, and an SMS-enabled device. Follow the instructions below to install the SCE Plugin adapter for the Identity Provider. The variable <PF-install> used in the instructions corresponds to the PingFederate installation path. In this build the PingFederate installation path was C:\pingfederate-7.3.0.

1. Log on to the server that hosts the PingFederate service for the Identity provider.
2. Download the SCE Plugin adapter jar file (e.g. sce-adapters-pingfederate-aa.1.1.jar) to the local PingFederate server.
3. Copy the jar file to <PF-install>/server/default/deploy
4. From the adapter dist/conf/template folder, copy all .html files to
   <PF-install>/server/default/conf/template
5. From the adapter dist/conf/template/assets folder, copy the aa folder to
   <PF-install>/server/default/conf/template/assets
6. From the adapter dist/data/adapter-config folder, copy the aa folder to
   <PF-install>/server/default/data/adapter-config
7. From the adapter dist/lib folder, copy all .jar files to
   <PF-install>/server/default/lib

2.10 Configure PingFederate-IdP

Follow the instructions in the subsections below to configure PingFederate as the Federation Server for the Identity Provider.

1. Launch your browser and go to: https://<DNS_NAME>:9999/pingfederate/app.
2. Replace DNS_NAME with the fully qualified name of the Identity Provider’s PingFederate server (e.g. https://idp.abac.test:9999/pingfederate/app).
3. Log on to the PingFederate app using the credentials you configured during installation.
2.10.1 Configure SAML Protocol

1. On the Main menu under System Settings, click Server Settings.

2. Click the Roles and Protocols tab. Select Enable Identity Provider (IdP) role and support the following.

3. Select SAML 2.0.

4. Click Save.
### 2.10.2 Create Data Store for Microsoft AD

1. On the **Main** menu under **System Settings**, click **Data Stores**.

![PingFederate](image1.png)

2. Select **LDAP**.

![PingFederate](image2.png)

3. Click **Next**.

4. Enter the **Hostname** where the Microsoft AD is hosted (e.g. `activedirectory.abac.test`).

5. For the **LDAP Type**, select **Active Directory**.

6. Enter the **User DN** created in section 2.4.1, Create the LDAP User for Federated Authentication (e.g. `CN=LDAP User, CN=Users, DC=ABAC, DC=Test`).

7. Enter the password associated with the LDAP User DN. Select the option to use LDAPS.

8. Click **Next**. Then, click **Save** on the **Summary** screen.
2.10.3 Create Credential Validator for Microsoft AD

1. On the **Main** menu under Authentication, click Password Credential Validators.

2. Click **Create New Instance**.

3. Enter a unique **Instance Name** you would like to use to refer to this configuration (e.g. AD username password).

4. Enter a unique **Instance Id** (typically the same as the **Instance Name**) without any spaces.

5. For **Type** select LDAP Username Password Credential Validator.
6. Click Next.

7. For the LDAP DATASTORE select the Active Directory data store you created earlier (e.g. activedirectory.abac.test).

8. Enter the SEARCH BASE (i.e. location in the directory where the LDAP search begins) for your Microsoft AD LDAP directory (e.g. DC=ABAC,DC=TEST).

9. Enter the SEARCH FILTER (e.g. sAMAccountName=${username}). The SEARCH FILTER allows Ping to search the LDAP directory, looking for a match where the attribute named sAMAccountName matches the username value passed from the PingIdentity server.

10. Click Next.
You should see two attributes listed under **CORE CONTRACT, DN, and username**.

11. Click **Next**.

You should see a summary page.

12. Click **Done**.

You should see a list of the credential validator instances, including the newly added validator (e.g. **AD username password**).
13. Click Save to complete configuration of the credential validator.

2.10.4 Create IdP Adapter for Authentication with Microsoft AD via Web Browser Form

The IdP Adapter created in this section is the logical component PingFederate uses to authenticate a user with Microsoft AD via a web browser login page.

1. On the Main menu under Application Integration Settings, click Adapters.

2. Click Create New Instance.

3. In Instance Name, enter a unique name for the instance. The name will be used to refer to this configuration (e.g. AD HTML forms).

4. Enter a unique Instance Id (typically the same as the instance name) without any spaces. For Type select HTML Form IdP Adapter.
5. Click Next.

6. Under **PASSWORD CREDENTIAL VALIDATOR INSTANCE**, click on the **Add a new row to Credential Validator's hyperlink**. This will add a new selection box under the **PASSWORD CREDENTIAL VALIDATOR INSTANCE** with the value of **-Select One-** in it. In that new box, select the credential validator for Microsoft AD that was created in an earlier section (e.g. AD username password).

7. Under **PASSWORD CREDENTIAL VALIDATOR INSTANCE** click the **Update hyperlink** on the right side of the page. This will cause the selection box to turn grey.
8. Click Next. Then, click **Next** again to bypass the Extended Contract screen.

9. On the Adapter Attributes screen, select the **PSEUDONYM** check box in the **username** row.

10. Click **Next**. On the Summary screen click **Done**.
11. Click **Save** to complete configuration of the new adapter.

### 2.10.5 Create IdP Adapter for Two-factor Authentication with RSA AA

The IdP Adapter created in this section is the logical component PingFederate uses to authenticate a user with RSA AA using a second factor.

1. On the **Main** menu under Application Integration Settings, click Adapters.
2. On the **Manage IdP Adapters** screen, click **Create New Instance**.
3. On the **Type** screen, enter an **Instance Name** and **Instance ID**.
4. Set the following settings on the Adapter Type page before clicking **Next**:
   a. **Instance Name**: [Instance Name]
   b. **Instance ID**: [Instance ID]
   c. **Type**: RSA Adaptive Authentication Adapter 2.0
   d. **Class Name**: com.thescegroup.adapters.aa.pingfederate.AdaptiveAuthenticationAdapter
   e. **Parent Instance**: None
Chapter 2. Setting up the Identity Provider

5. On the **IdP Adapter** configuration page, click **Show Advanced Fields** and input the following parameters while leaving the rest as default, before clicking **Next**:

   a. **AA Web Service URL**:
      
      http://<RSA Server DNS>:8080/AdaptiveAuthentication/services/AdaptiveAuthentication
   
   b. **AA Web Service Username**: [username]
   
   c. **AA Web Service Password**: [password]

   **Note**: The credentials must match on the RSA server.

6. On the **Extended Contract** screen, type **transactionid** (all lowercase). Then, click **Add**. By default, **username** should already be listed under **Core Contract**.
7. Click Next.

8. On the Authentication Context screen, select SecureRemotePassword as the fixed value for authentication. This value will be included in the SAML assertion. Click Next.

9. On the Adapter Attributes screen, select username as the Pseudonym. Click Next.

10. On the Summary screen, verify the information is correct and click Done.

11. On the Manager IdP Adapter Instances screen, click Save to complete the Adapter configuration.
2.10.6 Create Composite IdP Adapter Integrating Microsoft AD and RSA AA

The IdP Adapter created in this section is composite adapter that integrates the two previously created adapters for Microsoft AD and RSA AA. When a user is directed to the PingFederate IdP server, the user will see a web form where they can enter their Microsoft AD credentials. Following authentication with Microsoft AD, PingFederate will initiate the second factor authentication with an SCE Plugin. The SCE Plugin will then present the user with a request for the second factor.

1. On the **Main** menu under **Application Integration Settings**, click **Adapters**.
2. On the **Manage IdP Adapters** screen, click **Create New Instance**.
3. Enter a unique **Instance Name** you would like to use to refer to this configuration (e.g. RSA Multifactor).
4. Enter a unique **Instance Id** (typically the same as the **Instance Name**) without any spaces.
5. For **Type** select **Composite Adapter**.

6. Click **Next**.
7. On the **IdP Adapter** screen, under **ADAPTER INSTANCE**, click on the Add a new row to 'Adapters' hyperlink. This will add a new selection box under the **ADAPTER INSTANCE** with the value of **-Select One-** into the box. In that new box, select the adapter instance for html forms with Microsoft AD that was created in an earlier section (e.g. **AD HTML forms**).
8. Under **ADAPTER INSTANCE** click the **Update** hyperlink on the right side of the page. This will cause the selection box to turn grey.
9. Repeat the previous steps to add another row to Adapters using the hyperlink on the right side of the page. This time select the AdaptiveAuthentication adapter in the selection box. When complete the IdP Adapter screen will look similar to the screenshot below, with two adapters configured under ADAPTER INSTANCE.

10. Under TARGET ADAPTER, click on the Add a new row to 'Input User Id Mapping' hyperlink. This will add a new selection box under the TARGET ADAPTER with the value of -Select One- in the box.

11. In that new box, select the adapter instance for the RSA authentication that was created in an earlier section (e.g. AdaptiveAuthentication).

12. In the new USER ID SELECTION box, select username.

13. Under TARGET ADAPTER click the Update hyperlink on the right side of the page. This will cause the selection box to turn grey.
14. Click **Next**.

15. On the **Extended Contract** screen, enter the value **username** in the **EXTEND THE CONTRACT** field.

16. Click **Add**. Enter the value **transactionid** (all lowercase) in the **EXTEND THE CONTRACT** field.
17. Click **Add**. Then, click **Next**.

18. On the **Adapter Attributes** screen, in the **username** row, select the **PSEUDONYM** column.

19. Click **Next**. On the **Summary** screen, click **Done**.

20. Click **Save** to complete configuration of the new composite adapter.

### 2.10.7 Configure the Federation Connection to the Relying Party

This PingFederate SP Connection at the PingFederate-IdP will configure the SAML exchange with a server in the Relying Party’s environment. This connection will also enable a user to authenticate using the composite adapter created in the previous section.

1. On the **Main** menu under **SP CONNECTIONS**, click **Create New**.

2. On the **Connection Type** screen, make sure **Browser SSO Profiles** is selected.
Chapter 2. Setting up the Identity Provider

3. Click **Next**. On the Connection Options screen, make sure **Browser SSO** is selected.

4. Click **Next**.

5. On the Import Metadata screen, click **Browse** and select the metadata file that you exported from the Relying Party’s PingFederate server.
6. Click **Next**.

7. On the Metadata Summary screen, click **Next**.

8. On the General Info screen you should see some configuration information (e.g. **Base URL**) about the Relying Party that was taken from the metadata file that you selected earlier.

9. Click **Next**. On the Browser SSO screen, click **Configure Browser SSO**.

10. Select **IdP-Initiated SSO** and **SP-Initiated SSO**. Then, click **Next**.
Chapter 2. Setting up the Identity Provider

11. On the Assertion Lifetime screen, click **Next**.

12. On the Assertion Creation screen, click **Configure Assertion Creation**. This will bring up a sequence of sub screens starting with Identity Mapping.

13. On the Identity Mapping screen, select the **Standard** option.

14. Click **Next**. This will bring up the Attribute Contract screen.
15. Click **Next**.

16. On the Authentication Source Mapping screen, click **Map New Adapter Instance**. This will launch a sequence of sub-screens, beginning with the Adapter Instance screen.

17. On the Adapter Instance screen, select the composite adapter created in an earlier section (e.g. **RSA Multifactor**).
18. Click Next. On the Assertion Mapping screen, select **Use only the Adapter Contract values in the SAML assertion.**

19. Click Next.

20. On the Attribute Contract Fulfillment screen, for **SAML_SUBJECT**, select **Adapter** for the **SOURCE** field and **username** for the **VALUE** field.
21. Click **Next**.

22. Click **Next**.
23. Click **Done**. This will bring you back to the Authentication Source Mapping screen and you should see the composite adapter (e.g. **RSA Multifactor**) listed.

24. Click **Next**.

25. On the **Summary screen**, click **Done**. This will take you back to the Configure Assertion Creation screen.
26. Click **Next**.

27. On the **Protocol Settings** screen, click **Configure Protocol Settings**. This will launch a sequence of sub-screens, beginning with the **Assertion Consumer Service URL** screen.

28. On the **Assertion Consumer Service URL** screen, make sure that the **BINDING** field is set to **POST** and the **ENDPOINT URL** field is set to `/sp/ACS.saml2`. 
29. Click **Next**.

30. On the Allowable SAML Bindings screen, select **POST** and **Redirect**.

31. Click **Next**.

32. On the Signature Policy screen, select **Require AuthN requests to be signed when received via the POST or Redirect bindings**.

33. Click **Next**. On the Encryption Policy screen, select **The entire assertion**.
34. Click **Next**.

35. On the **Summary screen**, click **Done**.
Chapter 2. Setting up the Identity Provider

35. Click Next. On the Summary screen, click Done. This will take you back to the Browser SSO screen.

36. Click Next. On the Summary screen, click Done. This will take you back to the Protocol Settings screen.

37. Click Next.

38. On the Credentials screen, click Configure Credentials.

39. For the Signing Certificate field, select the certificate to be used to sign the SAML message.

40. Select the certificate that you configured for the server in an earlier section.

41. Select the Signing Algorithm for your environment (e.g. RSA SHA256).
42. Click Next.

43. Click Next.

44. On the Select XML Encryption Certificate screen, select the **Block Encryption Algorithm** (e.g. AES-128), and the **Key Transport Algorithm** (e.g. RSA-OAEP).

45. For the selection box above the **Manage Certificates** button, select the Relying Party’s public key certificate to be used to encrypt the message content.
46. Click **Next**.

47. On the **Summary** screen, click **Done**. This will take you back to the **Credentials** screen.
48. Click **Next**.

49. On the **Activation and Summary** screen, select **Active** for the **Connection Status** field.

50. Copy the Identity Provider’s **SSO Application Endpoint URL** (e.g. https://idp.abac.test:9031/idp/startSSO.ping?PartnerSpId=https://rp.abac.test:9031) to the clipboard and save it to a text file, because this URL will be used in the functional test section.

51. Click **Done**. This will take you to a screen that lists the connections for the server, including the new connection you just created. Click **Save** to complete the configuration.
### 2.11 Certificates

Once you have installed the various products for this ABAC build, you can replace the default self-signed certificates with certificates signed by a Certificate Authority. For our build, we used Symantec’s Managed PKI Service to sign our certificates using a local Certificate Authority. Certificates were used to support various exchanges that require encryption, such as digital signature, SAML message encryption, and encryption of TLS communications.

Although the detailed instructions of configuring certificates signed by a certificate authority vary by vendor product, this section describes the general process. For each certificate you perform the following high level steps:

1. Using the vendor product (e.g. PingFederate, Sharepoint), generate a certificate signing request on the server where you want to use the certificate. Save the signing request to a file.

2. Submit an enrollment request to your certificate authority. You will need to provide the signing request that was generated in step 1. This step is typically where you provide information such as the name of the server you intend to use the certificate on (e.g. `idp.abac.test`).

3. A representative at the certificate authority will examine the enrollment request and approve it. The representative will issue a certificate response signed with the certificate authority’s key. You can download the signed response. If you are using a certificate authority that is locally managed by your organization, you will also need to download the public key of the certificate authority because you will need to add this to the Trusted Certificate Authorities on each server and client that will be using the certificates.

4. Go back to the vendor product where you created the certificate signing request. If you are using a local certificate authority, you will first need to add the certificate authority’s public key to the list of Trusted Certificate Authorities.

5. Import the certificate file for your server that was signed by the certificate authority.

#### 2.11.1 Certificate Configuration PingFederate

In the PingFederate app, on the **Main** menu, under **Certificate Management**, click **Trusted CAs** to import the public key of your local certificate authority. If you are using a well-known, external, major certificate authority and that authority’s public key is already available in cacerts in the Java runtime, it is not necessary to import the same certificate into the PingFederate Trusted CA store.

- For SSL Server certificates follow the instructions in the link below. The applicable sections are **To create a new certificate**, **To create a certificate-authority signing request**, and **To import a certificate authority response**. Once you have imported a signed certificate response, you will need to activate the certificate on the PingFederate runtime server instance your applications are running on. Follow the instructions in the section **To activate a certificate**.
  
  [https://documentation.pingidentity.com/display/PF73/SSL+Server+Certificates](https://documentation.pingidentity.com/display/PF73/SSL+Server+Certificates)

- For digital signatures and performing encryption / decryption, follow the instructions in the link below. The applicable sections are the same as for SSL Server certificates.
  
  [https://documentation.pingidentity.com/display/PF73/Digital+Signing+and+Decryption+Keys+and+Certificates](https://documentation.pingidentity.com/display/PF73/Digital+Signing+and+Decryption+Keys+and+Certificates)
2.12 Functional Test of All Configurations for this Chapter

The instructions in this section will help perform an integrated test all of the configurations in this chapter. Using the browser and PingFederate, a user will log on and validate that the federated authentication to Microsoft AD and RSA AA are properly configured.

The test for this chapter was performed using the Mozilla Firefox browser and the SAML tracer Add-on, which enables examination of HTTPS POST and SAML messages.

1. Install the Firefox SAML tracer Add-on from the link below.
   https://addons.mozilla.org/en-Us/firefox/addon/saml-tracer/

2. Launch your Firebox browser and select SAML tracer from the Tools menu.

   ![Firefox Tools Menu]

   This will launch an empty SAML tracer window.

3. Minimize the SAML tracer window. The SAML tracer will automatically record the details of the HTTPS messages in the background.

4. Go back to the main browser window and navigate to the Identity Provider’s SSO Application Endpoint URL identified in the previous section (e.g. https://idp.abac.test:9031/idp/startSSO.ping?PartnerSpId=https://rp.abac.test:9031).

   **Expected Result:** You should see the PingFederate Sign On screen.

5. Enter the Username of the account created in Microsoft AD earlier in this chapter (e.g. lsmith).
6. Enter an invalid Password for the account. Do not enter the correct password.

7. Click **Sign On**.

   **Expected Result:** You should see an error message that states: *We didn't recognize the username or password you entered.*

8. Close the existing browser and launch a new browser.

9. Navigate to the Identity Provider’s SSO Application Endpoint URL again.

10. Enter the user name of the account created earlier in this chapter (e.g. *lsmith*). Then, enter the correct password.

11. Click **Sign On**.
**Expected Result:** You should see the two-factor RSA AA plugin screen. This screen prompts you to enter the SMS text validation code received by your mobile phone.

![Identity Verification via SMS](image1)

**Figure 2.3  Identity Verification via SMS**

12. Enter the SMS validation code received on your mobile phone and proceed. This will initiate a communication with the RSA AA server to validate the code that was entered.

**Expected Result:** The browser should redirect to the Relying Party’s Federation Server (e.g. `rp.abac.test`) and you should see an error message similar to the following screenshot.

![Confirmation Code Screen](image2)

**Figure 2.4  Confirmation Code Screen**
Chapter 2. Setting up the Identity Provider

13. Go back to the SAML tracer window. Scroll to the bottom of the list of messages in the upper pane. Click on the last message (e.g. POST https://rp.abac.test:9031/sp/ACS.saml2) that has a SAML icon associated with it. This will show the details of the POST message.

Expected Result: In the details page at the bottom, on the http tab, you should see that the browser sent a POST message to the Relying Party’s PingFederate server rp.abac.test. The HTTP response status code (identified on the line that begins with HTTP) should be a 500 Server Error.

14. Click on the SAML tab.

Expected Result: You should see the details of the SAML message, including the Issuer. The Issuer should be the Identity Provider’s Federation server, idp.abac.test.
Setting up Federated Authentication
Between the Relying Party and the Identity Provider

3.1 Introduction ........................................................................................................................ 66
3.2 Components .................................................................................................................... 66
3.3 Export Metadata from the Identity Provider ........................................................................ 67
3.4 Configure PingFederate-RP Connection to the PingFederate-IdP ..................................... 70
3.5 Functional Test of All Configurations for this Chapter ...................................................... 93
3.1 Introduction

In the previous chapter of this How-To Guide we demonstrated how to set up federated, SAML-based authentication at the Identity Provider (IdP). Before continuing with this chapter, it is necessary to have a working federation service that will represent the Identity Provider and can receive and issue SAML 2.0 request and responses. For instructions on how to set this up using PingFederate, please refer to chapter 2 of this guide.

In order to federate identities and attribute information between organizations a federation service must exist at both the Identity Provider and the Relying Party (RP). A trust relationship between these two services must then be instantiated to allow for identity and attribute requests and responses. In this chapter we configure an instance of PingFederate (henceforth called PingFederate-RP) at the Relying Party to act as a federation service and to redirect users to the PingFederate-IdP via a SAML request. We then configure the trust relationship and federated authentication between the PingFederate-RP and the PingFederate-IdP, allowing the SAML request to be processed by the Identity Provider and the subsequent return of a SAML response containing identity and attribute assertions.

If you follow the instructions in this chapter, you will be able to perform a functional test to verify the successful completion of the steps for installing, configuring, and integrating the components.

3.2 Components

Federated authentication between the Relying Party and the Identity Provider involves the following distinct components:

- **PingFederate-IdP**: A federation system or trust broker for the Identity Provider
- **PingFederate-RP**: Serves as the trust broker for SharePoint

3.2.1 PingFederate-IdP

Ping Identity PingFederate-IdP serves as a federation system or trust broker for the IdP. PingFederate-IdP provides initial user authentication and retrieval of user attributes to satisfy SAML requests from the RP. Once the user has been authenticated, PingFederate-IdP queries subject attributes from AD and environmental attributes from the RSA AA event log. PingFederate-IdP takes the name:value pairs of both the subject and environmental attributes and stores them in a SAML 2.0 token to be sent to the RP.

**PingFederate Usage Notes:**

- When using the PingFederate application to perform an administrative configuration, there is usually a sequence of screens that require user entry, ending with a summary page. Once you click **Done** on the summary page, you must also click **Save** on the following page to save the configurations. If you forget to click **Save**, you may inadvertently lose changes to the configuration.

- In the PingFederate application and associated documentation, the Relying Party is referred to as the Service Provider.
3. When using the PingFederate application to perform configuration, refer to the title of the tab with a small star icon to its left, to identify the item you are currently configuring. For example, if you navigated to the following screen, you would be on the IdP Adapter screen.

3.2.2 PingFederate-RP

Ping Identity PingFederate-RP serves as the trust broker for SharePoint. When the user requires authentication, PingFederate-RP redirects the user to the IdP via a SAML request to get the necessary assertions. Once authenticated, PingFederate-RP arranges for the browser's HTTPS content to have the proper information in proper format for acceptance at the target resource (SharePoint).

3.3 Export Metadata from the Identity Provider

Follow the instructions in this section to export a metadata file from the PingFederate-IdP.

1. Log on to the server that hosts the PingFederate service for the Identity Provider.

2. Launch your browser and navigate to the PingFederate application URL: https://<DNS_NAME>:9999/pingfederate/app.

3. Replace DNS_NAME with the fully qualified name of the Identity Provider's PingFederate server (e.g. https://idp.abac.test:9999/pingfederate/app). Log on to the PingFederate application using the credentials you configured during installation.

4. On the **Main Menu** under **Administrative Functions**, click **Metadata Export**.
5. On the Metadata Mode screen, select **Use a connection for metadata generation.**

6. Click **Next.** On the Connection Metadata screen, select the connection to the Relying Party that you configured in the previous chapter (e.g. https://rp.abac.test:9031). This should automatically populate some of the fields on the screen with information from the connection.
7. Click Next. On the Metadata Signing screen, if you plan to sign the metadata file that will be exported, select the certificate that will be use to sign the file.

8. Click Next. On the Export & Summary screen, you should see a summary of the options that were selected.
9. Click **Export**. This will create an export file that contains the metadata of the Identity Provider that you can download using the browser.

![Opening metadata.xml](image)

10. Copy the metadata file to the server that hosts the PingFederate service for the Relying Party.

### 3.4 Configure PingFederate-RP Connection to the PingFederate-IdP

Follow the instructions in this section to configure a PingFederate connection from the Relying Party to the Identity Provider.

1. Log on to the server that hosts the PingFederate service for the Relying Party.
2. Launch your browser and go to: `https://<DNS_NAME>:9999/pingfederate/app`.
3. Replace **DNS_NAME** with the fully qualified name of the Relying Party’s PingFederate server (e.g. `https://rp.abac.test:9999/pingfederate/app`). Log on to the PingFederate application using the credentials you configured in the previous installation section.
4. On the Main Menu under IDP CONNECTIONS, click Create New.

5. On the Connection Type screen, select Browser SSO Profiles.

6. Click Next.

7. On the Connection Options screen, make sure Browser SSO is selected.
8. Click **Next**.

9. On the Import Metadata screen, click **Browse** and select the metadata file that you exported from the Identity Provider's PingFederate server.

10. Click **Next**.
11. On the Metadata Summary screen, click **Next**. On the General Info screen you should see some configuration information (e.g. Base URL) about the Identity Provider that was taken from the metadata file that you selected.

12. Click **Next**.

13. On the Browser SSO screen, click **Configure Browser SSO**.
14. On the SAML Profiles screen, select **IdP-Initiated SSO** and **SP-Initiated SSO**.

15. Click **Next**.
16. On the User-Session Creation screen, click **Configure User-Session Creation**.

17. On the Identity Mapping screen, click **Next**.
18. On the Attribute Contract screen, click **Next**.

19. On the Target Session Mapping screen, click **Map New Connection Contract Mapping**.
20. On the Connection Mapping Contract screen, click **Manage Connection Mapping Contracts**.

![Manage Connection Mapping Contracts](image)

21. On the Manage Contracts screen, click **Create New Contract**.

22. On the Contract Info screen, enter the **Contract Name** (e.g. Sharepoint 2013).
23. Click Next.

24. Click Next.
25. On the Summary screen, click **Done**.

26. On the Manage Contracts screen, you should see the new contract listed. Click **Save**.

27. On the Connection Mapping Contract screen, for the **CONNECTION MAPPING CONTRACT** field select the name of the new contract that was created (e.g. **Sharepoint 2013**).
28. Click Next. On the Attribute Retrieval screen, select **Use only the attributes available in the SSO Assertion**.

29. Click Next. On the Contract Fulfillment screen, for the **SOURCE** field select **Assertion**. For the **VALUE** field, select **SAML_SUBJECT**.
30. Click **Next**.

31. On the Issuance Criteria screen, click **Next**.

32. On the Summary screen, click **Done**.
33. On the Target Session Mapping screen, you should see new contract (e.g. **Sharepoint 2013**) listed under the **CONNECTION MAPPING CONTRACT NAME** field.

34. Click **Next**.
35. Click **Done**.

36. On the User-Session Creation screen, click **Next**.
37. On the Protocol Settings screen, click **Configure Protocol Settings**. This will bring up a sequence of sub-screens.

38. On the SSO Service URLs screen, click **Next**.

39. On the Allowable SAML Bindings screen, select **POST** and select **Redirect**.
40. Click **Next**.

41. On the Default Target URL screen, click **Next**.

42. On the Signature Policy screen, make sure that the following are selected:
   
   a. **Specify additional signature requirements** and
   
   b. **Sign AuthN requests sent over POST and Redirect bindings**
43. Click Next. On the Encryption Policy screen, select:
   a. Allow encrypted SAML Assertions and SLO messages and
   b. The entire assertion

44. Click Next.
45. On the Summary screen, click **Done**.

46. On the Protocol Settings screen, click **Next**.

<table>
<thead>
<tr>
<th>Protocol Settings Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound SSO Bindings</td>
</tr>
<tr>
<td>Inbound Bindings</td>
</tr>
<tr>
<td>Signature Policy</td>
</tr>
<tr>
<td>Encryption Policy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocol Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SSO SERVICE URLS</strong></td>
</tr>
<tr>
<td>Endpoint</td>
</tr>
<tr>
<td>Endpoint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALLOWABLE SAML BINDINGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artifact</td>
</tr>
<tr>
<td>POST</td>
</tr>
<tr>
<td>Redirect</td>
</tr>
<tr>
<td>SOAP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEFAULT TARGET URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGNATURE POLICY</td>
</tr>
<tr>
<td>Sign AuthN requests own POST</td>
</tr>
<tr>
<td>Require digitally signed SAML</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ENCRYPTION POLICY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt Entire Assertion</td>
</tr>
<tr>
<td>Encrypt Name Identifier</td>
</tr>
<tr>
<td>Encrypt One or More Attributes</td>
</tr>
</tbody>
</table>
47. On the Summary screen, click **Done**.

![Browser SSO Configuration](image)

48. On the Browser SSO screen, click **Next**.

![Credentials screen](image)

49. On the Credentials screen, click **Configure Credentials**.
50. On the Digital Signature Settings screen, select:

a. **Signing Certificate for SAML messages and**

b. **Signing Algorithm**

51. Click **Next**.
52. On the Signature Verification Settings screen, click **Manage Signature Verification Settings**.

53. On the Trust Model screen, click **Next**.

54. On the Signature Verification Certificate screen, select the certificate to verify digital signatures.
55. Click **Next**.

56. On the **Summary** screen, click **Done**.

57. On the **Signature Verification** Settings screen, click **Next**.

58. On the **Select XML Decryption Key** screen, select the certificate associated with the private key that will decrypt messages from the Identity Provider.
59. Click **Next**.

60. On the **Summary** screen, click **Done**.

61. On the **Credentials** screen, click **Next**.
62. On the Activation and Summary screen, select **Active** for the **Connection Status** field.

![IdP Connection](image)

63. Copy the Relying Party's SSO Application Endpoint URL (e.g. https://rp.abac.test:9031/sp/startSSO.ping?PartnerIdpId=https://idp.abac.test:9031) to the clipboard and save it to a text file, because this URL will be used in the functional test section.

64. Click **Save** to save the configuration.

### 3.5 Functional Test of All Configurations for this Chapter

This section provides instructions to perform an integrated test all of the configurations in Chapter 2.

1. Using the browser and PingFederate, a user will log on at the Identity Provider, and then get redirected to the Relying Party.

   **Note:** This test is similar to the test in chapter 2, except this time the Relying Party has a destination endpoint connection that was configured in chapter 3, so the response code from the Relying Party's Federation server (e.g. rp.abac.test), should be an HTTP 200 status code.

2. Launch your browser and navigate to the Relying Party's SSO Application Endpoint URL identified in the previous section (e.g. https://rp.abac.test:9031/sp/startSSO.ping?PartnerIdpId=https://idp.abac.test:9031).

3. Launch the SAML tracer as in chapter 2 and minimize the tracer window.
**Expected Result:** You should see the PingFederate Sign On screen.

4. Enter the **Username** and **Password** of the account created in chapter 2 (e.g. lsmith) and click **Sign On**.

5. When the RSA Adaptive Authentication screen comes up, enter the SMS text validation code.

**Expected Result:** You should see the browser redirect to the Relying Party's Federation Server (e.g. rp.abac.test) and an error message similar to the message in the following screenshot.

6. Return to the SAML tracer window.

7. Scroll to the bottom of the list of message in the upper pane.
8. Click on the last message (e.g. POST https://rp.abac.test:9031/sp/ACS.saml2) that has a SAML icon associated with it. This will show the details of the POST message.

Expected Result: In the details page at the bottom, on the http tab, you should see that the browser sent a POST message to the Relying Party’s PingFederate server (e.g. rp.abac.test). The HTTP response status code (identified on the line that begins with “HTTP”) should be a 200 OK code.
4.1 Introduction

In previous sections of this How-To Guide, we installed several products to establish RP and IdP environments, their components, and the federation between them (Chapter 2 and Chapter 3). In this section of the How-To Guide we will illustrate how to install IIS (Internet Information Services 8), Microsoft SQL Server 2012, and Microsoft SharePoint Server 2013. Then, within SharePoint we will illustrate how to create a web application, configure the web application to run SSL, create a site collection, and create sub-sites.

In our build, we used ABAC policies and policy enforcement to protect RP resources like SharePoint sites and documents with the help of NextLabs products installed in subsequent How-To sections (Chapter 7 and Chapter 8).

4.1.1 Components Used in this How-To Guide

1. Internet Information Services (IIS) Manager - extensible web server created by Microsoft (formerly Internet Information Server) and is pre-installed in most Windows editions though is not active by default.

2. Microsoft SharePoint 2013 - Microsoft SharePoint is a web-based application within the Windows operating environment. Commonly, SharePoint is deployed as a document management system for intranet, extranet, or cloud repository purposes. SharePoint natively uses an RBAC authorization environment, but it also supports the use of attributes within the user transaction request, a capability Microsoft refers to as being "claims aware." SharePoint also allows for tagging data within its repository, which can be leveraged as object attributes.

3. Microsoft SQL Server 2012 - relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data.
### Required or Recommended Files, Hardware, and Software

<table>
<thead>
<tr>
<th>Component</th>
<th>Required Files</th>
<th>Required Other Software</th>
<th>Minimum Hardware Requirements</th>
<th>Recommended Hardware</th>
<th>Recommended or Minimum Operating System</th>
<th>Operating System or Other Software Used in this Build</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internet Information Services (IIS) 8</strong></td>
<td>Built-in component in Windows Server 2012 operating system (inactive by default) - Windows Server 2012 ISO</td>
<td>N/A</td>
<td>For the Windows 2012 Server OS: 512 MB RAM, 1.4 GHz 64-bit CPU, 32 GB hard disk; Gigabit Ethernet adapter</td>
<td>For the Windows 2012 Server OS: 800+ MB RAM, &gt;1.4 GHz 64-bit CPU, &gt;32 GB hard disk</td>
<td>Windows Server 2012 R2 Standard 64-bit</td>
<td>Windows Server 2012 R2 Standard 64-bit</td>
</tr>
<tr>
<td><strong>Microsoft SharePoint Server 2013</strong></td>
<td>SharePoint Server 2013 installation setup file or DVD</td>
<td>Microsoft SQL Server 2012; Microsoft SQL Server Management Studio; IIS 7.0 or 8.0 (Web Server Role, 8.0 required for Windows Server 2012)</td>
<td>12 GB RAM, 4 core, 64 bit CPU, 80 GB hard disk space for system drive</td>
<td>8+ GB RAM, 4+core 64-bit CPU, &gt;80 GB hard disk</td>
<td>The 64-bit edition of Windows Server 2008 R2 Service Pack 1 (SP1) Standard, Enterprise, or Datacenter or the 64-bit edition of Windows Server 2012 Standard or Datacenter</td>
<td>Windows Server 2012 R2 Standard 64-bit</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server 2012</strong></td>
<td>SQL Server 2012 setup file or DVD</td>
<td>.NET 4.0 Framework (SQL Server installs .NET 4.0 during the feature installation step.)</td>
<td>1GB RAM, 1.4GHz CPU, 6 GB of hard-disk space</td>
<td>4 GB RAM (should be increased as database size increases to ensure optimal performance), &gt;2.0 GHz CPU, 6 GH of hard-disk space</td>
<td>Windows Server 2008 R2 or Windows Server 2012, Windows 8.1, Windows 8, Windows 7 SP1, Windows Vista SP2</td>
<td>Windows Server 2012 R2 Standard 64-bit</td>
</tr>
</tbody>
</table>
4.2 Installation of Required Components

4.2.1 Installing SQL Server 2012

1. On the server where SQL Server 2012 is going to be installed, follow the steps from this link to install SQL Server 2012:


   a. Note: in our build, this SQL Server instance is leveraged by SharePoint Server 2013 and by the NextLabs ABAC policy definition, deployment, and enforcement components. Two of these NextLabs components are also installed on the same server as SQL Server 2012 (Chapter 7). In our build we call this server SQLServer.

      i. It is generally recommended by Microsoft regarding SharePoint Server and NextLabs regarding Control Center that the SQL Server be installed on a separate, dedicated server, which is why we chose that deployment in our build.

4.2.2 Installing IIS 8.0 on the SharePoint Server

1. On the separate server where SharePoint Server 2013 is going to be installed, follow the steps from this link to install IIS 8.0 (if not already installed; required for SharePoint Server 2013):

http://www.iis.net/learn/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012

   a. Note: in our build we call this the SharePoint Server.

4.2.3 Installing Microsoft SharePoint Server 2013

1. On the separate server where SharePoint Server 2013 is going to be installed, follow the steps from this link to install SharePoint Server 2013:


   a. Note: in our build we call this the SharePoint Server (same as step 2.2).

4.3 Creating the Web Application (IIS site) in SharePoint

1. On the SharePoint Server, open a web browser.

2. In the URL address bar of the browser, enter the address for Central Administration and click Enter or Go: http://sharepoint:4444/default.aspx
3. From the Central Administration page, click on **Application Management**.

4. On the Application Management Page, under the Web Applications section, click on **Manage web applications**.
5. From the left-most end of the Web Applications ribbon menu click on **New**.

![Web Applications ribbon menu](image)

6. In the Create New Web Application window that automatically opens, in the IIS Web Site section, do the following steps to choose the web application’s basic IIS configuration:

   a. Leave the radio button for **Create a new IIS web site** chosen (default).
   
   b. Leave the default **Name** or change the **Name** to something more memorable to you.
   
   c. Leave the default **Port** displayed or change the **Port** number to one that makes sense for your environment.

   ![Create New Web Application](image)

   d. Leave the **Host Header** blank and keep the default **Path**.

7. Further down in the Create New Web Application window, in the Security Configuration section, do the following steps to configure the web application to run SSL:

   a. Under **Allow Anonymous** leave the **No** radio button chosen (default).
85. Under **Use Secure Sockets Layer (SSL)**, click Yes.

86. Further down in the Create New Web Application window, in the Claims Authentication Types section, do the following steps to enable Windows Authentication (as illustrated):
   
a. Click on Enable Windows Authentication

88. b. Click on Integrated Windows authentication

90. 9. Further down in the Create New Web Application window, in the Claims Authentication Types section, note that there is a **Trusted Identity provider** section. Do not select this
option now, but later in our build and in other chapters there will be steps for setting up the federated logon.

Create New Web Application

any application pool account and with the default domain configuration.

Basic authentication method passes users’ credentials over a network in an unencrypted form. If you select this option, ensure that Secure Sockets Layer (SSL) is enabled.

ASP.NET membership and role provider are used to enable Forms Based Authentication (FBA) for this Web application. After you create an FBA Web application, additional configuration is required.

Trusted Identity Provider Authentication enables

10. Further down in the Create New Web Application window, in the Sign In Page URL section, leave the Default Sign In Page radio button chosen (default).

Create New Web Application

Sign In URL

When Claims Based Authentication types are enabled, a URL for redirecting the user to the Sign In page is required.

Learn about Sign In page redirection URL

11. Further down in the Create New Web Application window, in the Public URL section, change the URL or keep the default URL:

Create New Web Application

Public URL

The public URL is the domain name for all sites that users will access in this SharePoint Web application. This URL domain will be used in all links shown on pages within the web application. By default, it is set to the current servername and port.

https://SHAREPOINT:6454

12. Further down in the Create New Web Application window, in the Application Pool section, leave the default values:

a. Leave the radio button for Create new application pool chosen.

b. Note that the Configurable button is already chosen to select an existing security account for the new application pool, an account called SharePointAdmin in this build.
i. If you do not already have a managed account for this purpose, click on the Register new managed account link and follow the prompts to create one.

13. Further down in the Create New Web Application window, in the Database Name and Authentication section, leave the following fields filled in with the default information or enter your own manually:
   a. IP Address of the Database Server. In our build the separate, dedicated SQL Server IP address is 10.33.7.210
   b. Database name

14. Further down in the Create New Web Application window, in the Failover Server section, leave the Failover Database Server field blank.
15. Further down in the Create New Web Application window, in Service Application Connections, leave the default checkbox for **User Profile Service Application** checked.

16. Further down in the Create New Application window, in Customer Experience Improvement Program, either keep the **Enable Customer Experience Improvement Program** radio button for **No** chosen, or click on **Yes**.

17. At the bottom of the Create New Application window click **OK** to finish the web application creation process.

18. Wait for the new web application to be created.
19. In the Application Created window, click OK.

Application Created

The Microsoft SharePoint Foundation Web application has been created.

If this is the first time that you have used this application pool with a SharePoint Web application, you must wait until the Internet Information Services (IIS) Web site has been created on all servers. By default, no new SharePoint site collections are created with the Web application. If you have just created a Forms Based Authentication (FBA) Web application, then before creating a new site collection, you will need to perform some additional configuration steps.

Learn about how to configure a Web application for FBA.

Once you are finished, to create a new site collection, go to the Create Site Collection page.

20. Back on the Web Applications page, verify that your new SharePoint web application is listed ("SharePoint - 6454" from this example).
21. In another browser window, navigate to your new web application (e.g., https://sharepoint:6454/). Until the SSL certificate is installed as seen in the following section, you will receive this error.

![Secure Connection Failed](image)

### 4.4 Creating and installing SSL certificate

For a protected lab environment it is possible to use self-signed certificates, however for production network deployments it is generally recommended to use certificates signed by a Certificate Authority. Instructions related to both approaches are included in this section.

#### 4.4.1 Self-Signed Certificates

##### 4.4.1.1 Creating a Self-Signed Certificate on IIS 8

1. On the SharePoint Server, click on the Windows icon in the bottom left corner of your screen.
2. Begin typing IIS.
3. When the Internet Information Services (IIS) Manager appears, click on it.
4. Click on the **SharePoint Instance** to see its Features.

5. Scroll down and double-click on **Server Certificates**.

6. In the Server Certificates window, you will see any certificates that already exist.
7. In the Actions panel on the right side of the IIS Manager window, next to the Server Certificates window, click on Create Self-Signed Certificate.

8. In the Create Self-Signed Certificate window, Specify a friendly name for the certificate and Select a certificate store for the new certificate, then click OK.

4.4.1.2 Importing Self-Signed Certificate to SharePoint Certificate Store

1. After creating the self-signed certificate and clicking OK in the previous sub-section, you will see your new certificate.
2. Double-click on the new certificate.

3. In the Details tab of the Certificate window, click on Copy to File.
4. In the Certificate Export Wizard window that opens, click **Next**.

5. In the Certificate Export Wizard window on the Export Private Key screen, keep the selection **No, do not export the private key** and click **Next**.
6. In the Certificate Export Wizard window on the Export File Format screen, select the format you want to use (DER in this example), then click Next.

7. In the Certificate Export Wizard window on the File to Export screen, type in the certificate file name and click Next.
8. In the Certificate Export Window on the Completing the Certificate Export Wizard screen, click **Finish**.

![Certificate Export Wizard](image1)

9. In another Certificate Export Wizard window that automatically opens, you will see that the export was successful. Click **OK**.

![Certificate Export Wizard](image2)

### 4.4.1.3 Add the Self Signed Certificate to Trust management in Central Administration

1. Click on the Windows icon at the bottom left corner of your screen.
2. Begin typing the words: manage computer certificates.
3. Click on the Manage Computer Certificates icon.
4. In the certlm window, right-click on the SharePoint node, hover over All Tasks, then click Import.

5. In the Certificate Import Wizard window that opens, click Next.
6. In the Certificate Import Wizard window, on the File to Import screen, click **Browse** to find the self-signed certificate we created in the previous sub-section.

7. In the File Explorer window that opens automatically, click through location folders to find the self-signed certificate we created in the previous sub-section (example from this build: `C:/Windows/System32`).

8. Find the certificate and click to select it; then click **Open**.
9. Back at the Certificate Import Wizard, on the File to Import screen, the location of the self-signed certificate will be in the **File name** field. Click **Next**.

10. In the Certificate Import Wizard window on the Certificate Store screen, leave the default radio button for **Place all certificates in the following store chosen**. The **Certificate store** field should be set to SharePoint. Click **Next**.
11. In the Certificate Import Wizard window, click Finish.

12. In the Certificate Import Wizard window that automatically opens, you will see a message that the import was successful. Click OK.
13. In the certlm window, double-click on **Certificates** under the SharePoint node. The new self-signed certificate you created will be listed there.

14. Open **File Explorer** and click through locations to reach the location of your self-signed certificate (from this example: C:/Windows/System32/).

15. Right-click on the **self-signed certificate** and click on **Copy** or left-click on the self-signed certificate and press the keys Ctrl+C.

16. Right-click on your **Desktop** and click **Paste**, or left-click on your Desktop and press the keys Ctrl+V to save a copy of the certificate in an accessible location.

17. To Manage Trust via Central Administration, do the following steps: Open a **browser**.
18. In the **URL address bar** of the browser, enter the address for Central Administration and click Enter or Go: `http://sharepoint:44444/default.aspx`

19. From the Central Administration page, click on **Security** in the left-hand menu.

20. From the Security page, under the General Security section, click on **Manage Trust**.
21. Under the Trust Relationships tab of the Manage Trust page, click **New**.

![Image of Trust Relationships](image1.png)

22. In the Establish Trust Relationship window that opens automatically, enter the **Name** for the trust relationship being created, then click **Browse** to find the certificate created in previous sub-sections.

![Image of Establish Trust Relationship](image2.png)
23. In the Choose File to Upload window that opens automatically, navigate to the copy of your certificate from section 4.4.1.3 (e.g., Desktop). Click on the certificate so its name automatically fills the File name field at the bottom of the window, then click Open.

24. In the Establish Trust Relationship window, the certificate's location will be automatically entered as the Root Authority Certificate.
25. In the Establish Trust Relationship window, scroll down leaving the remaining fields empty, and click OK.

26. Your new trust relationship will be listed under the Trust Relationships tab.

4.4.1.4 Configure IIS Binding for the Self-Signed Certificate

1. Click on the Windows icon in the bottom left corner of your screen.

2. Begin typing IIS.
3. When the Internet Information Services (IIS) Manager appears, click on it.

4. On the left-hand side of the IIS Manager window, click on the SharePoint web application created in previous steps, then click Bindings in the Actions pane on the right.

5. In the Site Bindings window that opens, look for a binding type of https.
   a. If a binding type of https does not exist, click on Add.
   b. If a binding type of https does already exist, click on it, then click Edit.
6. In the Edit Site Binding window next to the SSL certificate field, click Select.

7. In the Select Certificate window, click on the certificate created in previous steps and click OK.
8. In the Edit Site Binding window, verify that your SSL certificate is listed, then click OK.

9. In the Site Bindings window, click Close.

4.4.2 Certificates Signed by Local or Online Certificate Authority

Instead of using self-signed certificates which can be used in protected lab environments, it is recommended that you use certificates signed by a Certificate Authority. For our build, we used Symantec's Managed PKI Service to sign our certificates using a local Certificate Authority. Certificates were used to support various exchanges that require encryption, such as digital signature, SAML message encryption, and encryption of TLS communications.

Although the detailed instructions of configuring certificates signed by a certificate authority vary by vendor product, the general process is described below. For each certificate you perform the following high level steps:

1. Using the vendor product (e.g., SharePoint), generate a certificate signing request on the server where you want to use the certificate. Save the signing request to a file.

2. Submit an enrollment request to your certificate authority. You will need to provide the signing request that was generated in step 1. This step is typically where you provide
information such as the name of the server on which you intend to use the certificate (e.g., "sharepoint.abac.test").

3. A representative at the certificate authority will examine the enrollment request and approve it. The representative will issue a certificate response signed with the certificate authority’s key. You can download the signed response. If you are using a certificate authority that is locally managed by your organization, you will also need to download the public key of the certificate authority because you will need to add this to the Trusted Certificate Authorities on each server and client that will be using the certificates.

4. Go back to the vendor product where you created the certificate signing request. If you are using a local certificate authority, you will first need to add the certificate authority’s public key to the list of Trusted Certificate Authorities.

5. Import the certificate file for your server that was signed by the certificate authority.

4.4.2.1 Generating a Certificate Signing Request (CSR)

1. Log into the server where SharePoint Server 2013 is installed (e.g., SharePoint Server in our build).

2. Click on the Windows icon in the bottom left corner of your screen.

3. Begin typing IIS.

4. When the Internet Information Services (IIS) Manager appears, click on it.

5. In the left-hand Connections column, left-click on your SharePoint instance.
6. Scroll down in the SharePoint Home pane and left-click on **Server Certificates**.

7. In the right-hand Actions column, click on **Open Feature**.
8. In the Server Certificates pane, in the right-hand Actions column, click on **Create Certificate Request**.

9. In the Distinguished Name Properties window that opens automatically, enter your organizational information and click **Next**.
10. In the Cryptographic Service Provider Properties window that opens automatically, choose the Cryptographic service provider and a Bit length, then click Next.

11. On the File Name screen, browse to the location where you would like to save this certificate or type in the path, including a name for your certificate ending in ".txt," then click Finish.
4.4.2.2 Installing the new signed SSL Certificate

When the new signed SSL Certificate is available either from a local or online Certificate Authority, install the certificate using the instructions in this section.

1. Log onto the SharePoint Server and save the SSL certificate resulting from the CSR in section 4.4.1.2.

2. Click on the Windows icon in the bottom left corner of your screen.

3. Begin typing IIS.

4. When the Internet Information Services (IIS) Manager appears, click on it.

5. In the left-hand Connections column, left-click on your SharePoint instance.

6. Scroll down in the SharePoint Home pane and left-click on Server Certificates.
7. In the right-hand Actions column, click on **Open Feature**.

8. In the Server Certificates pane, in the right-hand Actions column, click on **Complete Certificate Request**.

9. In the Complete Certificate Request wizard on the Specify Certificate Authority Response screen, browse to the location of the new SSL certificate generated from your CSR or type in
its location, enter a friendly name, and choose a certificate store from the drop-down menu. Click OK.

4.4.2.3 Configure the CA-Signed Certificate

Follow the steps listed in section 4.4.1.4 to configure IIS Binding for the new SSL certificate signed by a local or online Certificate Authority. You can choose port 443 or any other available port if you prefer to use a non-standard port for SSL traffic.

4.5 Creating a site collection

1. On the SharePoint Server, open a web browser.
2. In the URL address bar of the browser, enter the address for Central Administration and click Enter or Go: http://sharepoint:44444/default.aspx
3. From the Central Administration page, in the Application Management section, click on Create site collections.

4. On the Create Site Collection page, do the following:
   a. Verify that the web application under consideration is the one chosen.
   b. Enter a Title (required) and Description (optional).
   c. Choose the web site address you prefer for your site (in this build, https://sharepoint:6454/).
5. In the browser, scroll down to the Template Selection area and Primary Site Collection Administrator area of the Create Site Selection page and do the following:
   a. Choose the **version** and **template** (e.g., 2013 Team Site)
   b. In the **User name** field, under the Primary Site Collection Administrator area, type in the name of your SharePoint Administrator account and click on the **Name check** icon. If the name is found, it will not give a warning and the name will be underlined.
      i. Alternatively, you can look up users by name using the address book people picker mechanism next to the user name text field.
   c. In the **User name** field under the Primary Site Collection Administrator area, type in the name of a secondary administrator if you so choose.
      i. Alternatively, you can look up users by name using the address book people picker mechanism next to the user name text field.

6. Scroll down in the browser to the Quota Template area of the Create Site Collection page. Leave the default choice **No Quota** chosen. Click **OK**.
7. Wait for the Site Collection to successfully complete.

Working on it...

This shouldn’t take long.

8. In the browser, on the page that indicates a new top-level site was created successfully, click **OK**.

9. Open a browser and navigate to the URL for your new web application (e.g., https://sharepoint:6454)
a. You may see a warning first because of the self-signing certificate.

b. In the browser window, click on I Understand the Risks, then Add Exception.

c. In the Add Security Exception window, click on Confirm Security Exception.
10. In the Authentication Required window that opens automatically, enter the administrator account **User Name** and **Password**, then click **OK**.

11. Upon verification that the login was a success, you will see default site contents.
4.6 Creating new sub-sites

1. After logging into your site, in your browser window click the gear symbol next to the Administrator login area, then click on Site Contents.

2. In the browser window, the Site Contents page will open.
3. In the browser window, scroll down to the Subsites area and click the **plus sign button** next to new subsite.

4. In the browser window on the New SharePoint Site screen, do the following:
   a. Enter **Title** (required) and **Description** (optional).
   b. Enter a **URL name**.
c. Select a template.

5. In your browser, scroll down and do the following:
   a. Choose **User Permissions** (in our build, we left the Use same permissions as parent site radio button selected).
b. Choose your **Navigation** and **Navigation Inheritance** settings.

6. In the browser, scroll down and click **Create**.

7. Your new subsite will open in the browser.

8. Return to the homepage URL [https://sharepoint:6454](https://sharepoint:6454) and repeat the steps from section 4.6 to create other subsites of interest.
5 Set up Federated Authentication at the Relying Party’s SharePoint

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>144</td>
</tr>
<tr>
<td>5.2</td>
<td>Usage Notes on PingFederate</td>
<td>144</td>
</tr>
<tr>
<td>5.3</td>
<td>Configure a SharePoint Federated Logon Provider</td>
<td>145</td>
</tr>
<tr>
<td>5.4</td>
<td>Configure the PingFederate-RP Connection to SharePoint</td>
<td>157</td>
</tr>
<tr>
<td>5.5</td>
<td>Functional Test of All Configurations for This Chapter</td>
<td>171</td>
</tr>
<tr>
<td>5.6</td>
<td>Troubleshooting SharePoint Federated Authentication Problems</td>
<td>175</td>
</tr>
</tbody>
</table>
5.1 Introduction

In previous chapters of this How-To Guide we demonstrated how to set up set up federated authentication between the Relying Party and the Identity Provider and how to create the Relying Party’s SharePoint site. In this chapter we demonstrate how to set up federated authentication between the Relying Party’s SharePoint and the PingFederate-RP. Before continuing with this chapter implementers are required to have federation servers at both the Identity Provider and the Relying Party as well as a working SharePoint instance that is claims-aware. For this build we provide instructions for setting up these components in chapter 2, chapter 3, and chapter 4.

We will demonstrate how to set up a trusted logon provider for the Relying Party so that when a user requests access to a SharePoint site, the user will be redirected to the PingFederate-RP for authentication via WS-Federation. The Ping-Federate-RP will then forward the authentication request to the PingFederate-IdP. The PingFederate-IdP will present a logon page to the user. Once the user authenticates, the user will be redirected back to the original SharePoint site and will be able to access the site because they have a valid authentication token.

As you complete different steps in this chapter you will be able to verify the correctness or completeness of your component configuration and integration in functional test sub-sections. If you follow the instructions in this chapter, you will be able to perform a functional test to verify the successful completion of the steps for installing, configuring, and integrating the components.

5.2 Usage Notes on PingFederate

- When using the PingFederate application to perform an administrative configuration, there is usually a sequence of screens, ending with a summary page. Once you click Done on the summary page, you must also click Save on the following page to save the configurations. If you forget to click Save, you may inadvertently lose changes to the configuration.

- Ping identity refers to the Relying Party as the Service Provider in their PingFederate product and associated documentation.

- When using the PingFederate application to perform configuration, refer to the title of the tab with a small star icon to its left, to easily identify the item you are currently configuring. For example, if you navigated to the following screen, you would be on the IdP Adapter screen.
5.3 Configure a SharePoint Federated Logon Provider

Follow the instructions in this section to configure the federated logon provider at the Relying Party's SharePoint site. Once this configuration is complete, the user will see two authentication options when first attempting to access the SharePoint site. The first option is to log on using the default Windows Authentication. This option does not use federation. The second option is to use a federated logon.

In order to set up a federated logon, you will configure a trust relationship between the SharePoint server and the PingFederate-RP that will facilitate the federated logon. Once a user authenticates via a federated logon, the PingFederate-RP will cryptographically sign WS-Federation messages and send them to the SharePoint server. The PingFederate-RP must be configured as a trusted identity token Issuer in SharePoint, so that SharePoint will accept the messages sent by the PingFederate-RP and allow the user access to the SharePoint site.

5.3.1 Setting up the Certificate

Setting up a certificate involves creating the certificate at the from the Identity Provider, exporting the certificate, and importing it in the SharePoint site of the Relying Party.

1. Log on to the server that hosts the PingFederate service for the Relying Party.
2. Launch your browser and go to: https://<DNS_NAME>:9999/pingfederate/app.
3. Replace DNS_NAME with the fully qualified name of the Relying Party’s PingFederate server (e.g. https://rp.abac.test:9999/pingfederate/app).
4. Log on to the PingFederate application using the credentials you configured during installation.
5. On the **Main** menu, under **CERTIFICATE MANAGEMENT**, click **Digital Signing and XML**.

6. Locate the certificate that will be used to sign messages that will be sent to the SharePoint server. In the example screenshot above, this certificate has **CN** with the value **demo dsig new**.

7. Click on the **Export** link for this certificate in the **ACTION** column.
8. Select **Certificate Only** and click **Next**.

9. On the Export & Summary page, click the **Export** button on the left side of the page. Save the file to the hard drive and rename it to **federation.cer**.

10. Using the SharePoint administrator credentials, log on to the server that hosts SharePoint for the Relying Party.

11. Copy the **federation.cer** file to the desktop on the SharePoint server.

12. Click on the **Start** menu and navigate to the **SharePoint 2013 Products** group. Open the SharePoint 2013 Management Shell.
13. To verify that you placed the federation.cer file to the desktop, enter the following command into the Management Shell (using the correct path for your server).

dir c:\users\SharePointadmin\desktop\federation.cer

You should see information about the file such as the LastWriteTime.

14. Enter the following commands into the Management Shell to import the PingFederate-RP’s signing certificate (using the correct path for your server):

New-SPTrustedRootAuthority -Name "Federated Token Signing Cert" -Certificate $cert

SharePoint responds by displaying details about the imported certificate.
5.3.2 Configuring the Trusted Identity Token Issuer

To configure a new Trusted Identity Token Issuer, enter each of the commands displayed below the next paragraph into the Management Shell to configure a new Trusted Identity Token Issuer. Enter each command separately, and enter a Carriage Return after the command. If the command executed successfully, Management Shell will not provide any feedback. If an error occurs, Management Shell will display the error.

In the example commands below, the attribute **upn** is configured. You can replace **upn** with an attribute that is appropriate for your environment. The realm value (e.g. urn:SharePoint.abac.test) must be identical to the realm value configured in the Relying Party’s PingFederate Service Provider (SP) connection that will be configured later in this chapter. The signInURL should be configured with the PingFederate-RP WS-Federation URL (e.g. https://rp.abac.test:9031/idp/prp.wsf). In this example, the name given to this new token issuer in SharePoint is **Federated Logon from Identity Provider**. The issuer name will be displayed in SharePoint administration screens and to the end user on the Sign On screen.

```
$claimmap = New-SPClaimTypeMapping -IncomingClaimType "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn" -IncomingClaimTypeDisplayName "upn" -SameAsIncoming
$realm = "urn:SharePoint.abac.test"
$signInURL = https://rp.abac.test:9031/idp/prp.wsf
$ap = New-SPTrustedIdentityTokenIssuer -Name "Federated Logon from Identity Provider" -Description "Federated Logon" -realm $realm -ImportTrustCertificate $cert -ClaimsMappings $claimmap -SignInUrl $signInURL -IdentifierClaim $claimmap.InputClaimType
```
5.3.3 Configuring the Token Issuer as a Sign On Option

After configuring the new Trusted Identity Token Issuer, configure the new token issuer as a Sign On option for the SharePoint site.

1. Launch your browser and go the SharePoint central administration page (e.g. http://SharePoint.abac.test:44444/default.aspx).
2. Log on using the credentials of the SharePoint administrator.
3. In the Application Management group, click on Manage web applications.
4. Click on the web application that contains the SharePoint site you are managing (e.g. SharePoint - 80). SharePoint will highlight the web application row that you clicked on.
5. Click on the Authentication Providers button at the top of the page.
6. Click on the Default link in the Zone column.
7. On the Edit Authentication screen, scroll down to the Claims Authentication Types group. Select the Trusted Identity provider option.
8. Under the Trusted Identity provider checkbox, select the name of the new token issuer that was created using the Powershell commands (e.g. Federated Logon from Identity Provider).
9. Scroll to the bottom of the page and click **Save**.

### 5.3.4 Configuring the Access Control Rule on SharePoint

After configuring the token issuer as a Sign On option for SharePoint, configure the access control rule on the SharePoint site that is necessary for federated users to be able to access the site.

1. Log on to the Relying Party's SharePoint site (e.g. [https://SharePoint.abac.test](https://SharePoint.abac.test)) using the credentials of the SharePoint administrator.

2. Select **Windows Authentication** in the Sign On screen.

3. Click the gear icon at the top right corner of the page and select the **Site Settings** link.

4. On the Site Settings screen, in the Users and Permissions group, click **People and Groups**.
5. Under the Groups heading on the left pane, click on the HOME Members group.

6. Under the page title, click on the New link and select the Add Users option from the popup menu.

7. On the Share popup screen, enter Everyone in the text field. SharePoint will display a list box underneath the text field.

The list will contain multiple entries for the same value of Everyone. If you place your cursor over an entry in the list SharePoint will display details about the entry.
8. Locate the entry that is associated with All Users.

9. Click on the entry associated with All Users.

10. Click Share.

When you go back to the People and Groups screen, you should see Everyone listed for the Home Members group.

### 5.3.5 Functional Test of the Federated Logon at the Resource Provider

1. Launch a new browser window and go to the Relying Party's SharePoint site (e.g. https://SharePoint.abac.test).

**Expected Result:** You should see two logon options in the dropdown box. One of the options should be the name of the new trusted token issuer that was configured in the previous section (e.g. Federated Logon from Identity Provider).
Next you will verify that SharePoint is configured to read the upn attribute that was configured for the federated logon.

2. Launch your browser and go to the SharePoint central administration page (e.g. http://SharePoint.abac.test:44444/default.aspx).

3. Log on using the credentials of the SharePoint administrator.

4. In the Application Management group, click on Manage web applications.

5. Click on the web application that contains the SharePoint site you are managing (e.g. SharePoint - 80). SharePoint will highlight the web application row that you clicked on.
6. Click on the User Policy button.

7. Click Add Users.

8. Click Next.
9. On the Add Users screen, click the small browse icon (looks like a book) under the Users field.

**Expected Result:** On the Select People and Groups screen, you should see a grouping with the name of the trusted token issuer that was configured via Powershell (e.g. **Federated Logon from Identity Provider**). You should also see the **upn** attribute listed under that grouping.
5.4 Configure the PingFederate-RP Connection to SharePoint

Follow the instructions below to configure a PingFederate connection from the PingFederate-RP to the Relying Party's SharePoint.

1. Log on to the server that hosts the PingFederate service for the Relying Party.

2. Launch your browser and go to: https://<DNS_NAME>:9999/pingfederate/app. Replace DNS_NAME with the fully qualified name of the Relying Party's PingFederate server (e.g. https://rp.abac.test:9999/pingfederate/app). Log on to the PingFederate application using the credentials you configured during installation.

3. On the Main menu under SP CONNECTIONS, click Create New. On the Connection Type screen, select Browser SSO Profiles. For the Protocol field, select WS-Federation.
4. Click **Next**. On the Connection Options screen, select **Browser SSO**.

5. Click **Next**. On the General Info screen, for the Partner’s Realm field, enter the name of the Resource Provider’s (SharePoint) realm (e.g. urn:SharePoint.abac.test). Keep a copy of the realm name because it will be used in a configuration of SharePoint later in the guide.

6. Enter a unique name for this new PingFederate configuration in the **Connection Name** field. For the Base URL field, enter the root destination URL at the SharePoint site where the PingFederate will redirect a user once authenticated (e.g. https://SharePoint.abac.test).

7. Click **Next**.
8. On the Browser SSO screen, click **Configure Browser SSO**. On the Assertion Lifetime screen, enter a value of **20** for the **Minutes After** field.

9. Click **Next**.
10. On the Assertion Creation screen, click **Configure Assertion Creation**. On the Identity Mapping screen, select **User Principal Name**.

11. Click **Next**. On the Attribute Contract screen, below the **EXTEND THE CONTRACT FIELD**, enter `upn` in the text box. For the **ATTRIBUTE NAME FORMAT** select the `schemas.xmlsoap.org 2005 identity claims format`.

12. Click **Add**.
13. Click **Next**.

14. On the Authentication Source Mapping screen, click **Map New Connection Contract Mapping**. On the Connection Contract Mapping screen, for the **CONNECTION MAPPING CONTRACT** field, select the name of the contract with the Identity Provider that was configured in **chapter 3** (e.g. **SharePoint 2013**).
15. Click Next. On the Assertion Mapping screen, select **Use only the Connection Mapping Contract values in the SAML assertion**.

16. Click Next.
17. On the Attribute Contract Fulfillment screen, click **Next**.

18. On the Issuance Criteria screen, click **Next**.
19. On the Summary screen, click **Next**.

20. On the Authentication Source Mapping screen, click **Next**.

21. On the Summary screen, click **Done**.
22. On the Assertion Creation screen, click **Next**.

23. On the Protocol Settings screen, click **Configure Protocol Settings**.

24. On the Service URL screen, for the **Endpoint URL** field, enter the name of the destination URL at the Service Provider (SharePoint) site (e.g. /_trust/). When PingFederate completes the authentication process, the user will be sent to a destination URL. The destination URL is a combination of two configuration fields. The first is the **Base URL** that was configured earlier, and the second is the **Endpoint URL** on this screen. The **Endpoint URL** will be appended to the **Base URL**. An example is provided below.

**Base URL**: https://SharePoint.abac.test

**Endpoint URL**: /_trust/

After authentication, PingFederate will redirect to the destination:

https://SharePoint.abac.test/_trust/
25. Click **Next**.

26. On the Summary screen, click **Done**.
27. On the Protocol Settings screen, click **Next**.

28. On the Summary screen, click **Done**.
29. On the Browser SSO screen, click **Next**.

30. On the Credentials screen, click **Configure Credentials**.

31. On the Digital Signature Settings screen, select the **Signing Certificate** for SAML messages.
32. Click Next.

33. On the Summary screen, click Done.
34. On the Credentials screen, click Next.

35. On the Activation and Summary screen, select Active for the Connection Status field and click Save to complete the configuration.
5.5 Functional Test of All Configurations for This Chapter

The instructions in this section will perform an integrated test all of the configurations in this chapter.

1. Using the browser, you logon using an account that was created in Active Directory and validate that the complete federated authentication flow between SharePoint and the PingFederate servers at the Relying Party and Identity Provider operates successfully.

2. Launch your Firebox browser and select SAML tracer from the Tools menu.
   This will launch an empty SAML tracer window. Minimize the SAML tracer window. The SAML tracer will automatically record the details of the HTTPS messages in the background.

3. Go back to the main browser window and go to the Relying Party’s SharePoint site (e.g. https://SharePoint.abac.test).

4. Select the option to use the new trusted token issuer (e.g. Federated Logon from Identity Provider) that was configured in this chapter.

   **Expected Result:** Your browser should be redirected to the PingFederate-IdP and you should see the PingFederate Sign On screen. Examine the server name in the URL to ensure that it is the Identity Provider’s PingFederate server (e.g. idp.abac.test).
5. Enter the **Username** and **Password** of the Active Directory account created earlier in this guide (e.g. Ismith).

6. Click **Sign On**. On the RSA Adaptive Authentication screen, enter the SMS validation code received on your mobile phone. Click **Next**.

**Note:** Once authenticated at the Identity Provider, your browser should automatically redirect to the PingFederate-RP (e.g. *rp.abac.test*) and then to the Relying Party’s SharePoint (*SharePoint.abac.test*) site. Depending on the processing time of the servers in your environment, and other factors, it may take several seconds before your browser arrives back at the SharePoint site. The Identity Provider will redirect your browser to the PingFederate-RP first, and then the PingFederate-RP will redirect your browser to the SharePoint site, however you may not notice all of this activity if it happens quickly.

**Expected Result:** Go back to the SAML tracer window. Scroll down the list of messages at the top and ensure there is a POST message to the SharePoint server to the **_trust URL** (e.g. POST https://SharePoint.abac.test/_trust/).
7. Click on the POST message to the SharePoint _trust URL to bring up the details of the message in the bottom pane.

8. Click on the Parameters tab for the bottom pane.

9. Copy all of the content (beginning with the POST line) in the bottom page and paste it into a text editor such as Notepad. Turn on Word Wrap to make it easier to see all of the XML content.
10. Scroll down the SAML message and locate the AttributeStatement node and sub-nodes.

11. For the AttributeStatement node and sub-nodes, enter some carriage returns before each XML tag to make it easier to examine the data. The goal is to be able to easily examine the Attribute nodes within the AttributeStatement node.

**Expected Result:** Within the AttributeStatement node, there should be an Attribute sub-node. The Attribute sub-node should have an AttributeName value of upn. The AttributeNamespace value should be
http://schemas.xmlsoap.org/ws/2005/05/identity/claims. There should be an
AttributeValue sub-node and it should contain the account username (e.g. Ismith) that was
used to authenticate at the Identity Provider (e.g. <saml:AttributeValue>lsmith</saml:AttributeValue>.

**Expected Result:** Verify that the name (and case) of the attribute (noted by the
AttributeName) is identical to the name configured at the SharePoint using Powershell
earlier in this chapter. Verify that the AttributeNamespace is identical to the
IncomingClaimType option configured at the SharePoint using Powershell earlier in this
chapter. If the name or namespace of the attribute being passed to SharePoint does not
match with the SharePoint configuration, SharePoint will not allow access to the site, and
direct your browser back to the SharePoint Sign On screen.

12. If you verified that the name and namespace of the expected attribute match with the
SharePoint configuration and SharePoint does not direct your browser to the site home
page, follow the instructions in section 5.6, Troubleshooting SharePoint Federated
Authentication Problems, to determine the cause of the problem.

**Expected Result:** Go back to the main browser window. The SharePoint server should
present the site home page. You should see the account username of the user that
authenticated in the upper right corner of the page.

### 5.6 Troubleshooting SharePoint Federated Authentication Problems

If you encounter a situation where SharePoint is not allowing a federated user access to the
site, you may have a problem with the authentication configuration. A symptom that indicates
you have an authentication configuration problem is when a user successfully signs on at the
Identity Provider, then the user is redirected back to the SharePoint site, and instead of
displaying the site home page, SharePoint presents the SharePoint Sign On screen again. This
section describes how to determine the root cause of this type of authentication problem so
that the problem can be resolved.
**Note:** A SharePoint access control problem is a distinctly separate issue from authentication. A symptom of an access control problem is when the user received a message that states "This site has not been shared with you" upon successful authentication. Access control problems can be resolved by setting up SharePoint permissions on the People and Groups administration page, located in the Site Settings, Users and Permissions group.

Follow these instructions to troubleshoot federated authentication problems at the SharePoint site.

Before you configure diagnostic logging for the SharePoint site to determine the root cause of the authentication problem, check the following items first:

1. Verify that the Relying Party’s PingFederate Server and the Relying Party’s SharePoint Server synchronize their clocks from the same source. If both servers are on the same domain, they should be synchronized with the domain controller automatically. Log on to both servers and verify that the clocks display the same time.

2. Verify that the expiration time of the security token generated by the PingFederate Server is more than 10 minutes.

   SharePoint calculates the time length of its session using the formula:
   \[ \text{SharePointSessionTime} = \text{SecurityTokenLifeTime} - \text{LogonTokenCacheExpirationWindow}. \]

   *SecurityTokenLifeTime* is the length of time the token is valid, and this time is generated by the PingFederate server when it issues the token.

   By default the *SharePoint LogonTokenCacheExpirationWindow* is set to 10 minutes, therefore the *SecurityTokenLifeTime* must be greater than 10 in order to generate a *SharePointSessionTime* greater than zero.

   In our build we set the *SecurityTokenLifetime* to 20 minutes in the PingFederate configuration.

3. The expiration time of the security token can be set in the configuration of the SP Connection on the Relying Party’s PingFederate server. When you open the configuration for the SP Connection, click on the *Assertion Lifetime* link in the Browser SSO section. Enter a value for the *Minutes After* field that is greater than 10 (e.g. 20).
If you checked the items in the previous section and you are still encountering authentication problems, you will need to examine detailed authentication logs on the SharePoint server. Follow the instructions below to configure diagnostic logging on the SharePoint server and analyze the logs to determine the root of the authentication problem.

1. Perform the instructions at the following link to change the levels of ULS authentication logging on the SharePoint server. Make sure that you perform the instructions in the following two sections of the article:
   - To configure SharePoint 2013 for the maximum amount of user authentication logging
   - To find the failed authentication attempt manually


2. Once you configure the SharePoint diagnostic authentication logging, perform the sign on process to your SharePoint again to generate activity in the log.

   Tip: Since the SharePoint ULS log file contains many entries, it can be helpful to copy the file to another computer and analyze it offline.

3. Open a copy of the log file and scroll to the bottom of the file. The bottom of the log contains the most recent activity.

4. Starting at the bottom of the file, perform an upward search for the term authentication. Examine the entries that are labeled either Claims Authentication or Authentication Authorization.

5. Look at the details for each of these two types of authentication entries to look for clues regarding what the source of the problem could be. You may have to look through several entries in the file to understand the sequence of events.

We used this approach to troubleshoot an authentication problem in our lab. We found the following entry in the log file, that seemed as though it could be the source of the problem:

- security token '0e.t|federated logon from Identity Provider|lsmithc221cd9-23d7-4302-b029-ee81784754d2_Internet' is found in the local cache, but it is expired. Returning Null.

Two lines further down in the file, we found the following entry as well:

- Token Cache: Failed to find token for user '0e.t|federated logon from Identity Provider|lsmith' for cookie so signing out the user.

Based on the log file, we performed an Internet search for the term security token is found in the local cache, but it is expired. Returning Null. By researching various Internet blogs and forums, and performing additional analysis of the log file, we found a blog article on the PingIdentity website that described why the lifetime of the security token generated by the PingFederate-RP must be greater than 10 minutes when issuing a token for SharePoint. Once we updated the associated configuration on the PingFederate-RP, the authentication problem was resolved.

Identity Provider|Identity Provider
## Attribute Exchange Between the Identity Provider and Relying Party

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>180</td>
</tr>
<tr>
<td>6.2</td>
<td>Create Custom User Attributes in Microsoft AD</td>
<td>180</td>
</tr>
<tr>
<td>6.3</td>
<td>Configure PingFederate Servers to Pull User Attributes</td>
<td>193</td>
</tr>
<tr>
<td>6.4</td>
<td>Configure PingFederate-RP and SharePoint to Pass and Read Attributes</td>
<td>220</td>
</tr>
<tr>
<td>6.5</td>
<td>Configure the Claims Viewer Web Part at the SharePoint Site</td>
<td>231</td>
</tr>
<tr>
<td>6.6</td>
<td>Functional Test of All Configurations for this Chapter</td>
<td>237</td>
</tr>
</tbody>
</table>
6.1 Introduction

In previous chapters of this How-To Guide, we demonstrated foundational steps to building an ABAC solution:

- Configuring federated authentication at the PingFederate-IdP
- Configuring the SAML exchange between the PingFederate-Idp and PingFederate-RP
- Configuring the Relying Party’s SharePoint site
- Configuring the federated logon at the SharePoint site

Building upon that foundation, this chapter describes how to:

- Create custom attributes and set values for them in the Microsoft AD
- Configure the PingFederate-IdP to pull user and environmental attributes during authentication
- Configure the PingFederate-RP to pass the user and environmental attributes to the Relying Party’s SharePoint
- Configure SharePoint to load the user and environmental attributes passed from the PingFederate-RP into the web session

If you follow the instructions in this chapter, you will be able to perform a functional test to verify the successful completion of the steps for installing, configuring, and integrating the components.

6.2 Create Custom User Attributes in Microsoft AD

Follow the instructions in this section to create custom user attributes in the Microsoft AD schema. You will add a new attribute and add it to the user class. Microsoft AD user accounts inherit from the user class, therefore the new attribute will be available to all of the users in the domain.

6.2.1 Preparing the AD Schema for Creating New Custom Attributes

6.2.1.1 Backing up Your Directory before Making Schema Changes

Microsoft recommends that you backup your directory before making schema changes. Choose the names of your new custom attributes carefully, because the creation of a new attribute is a permanent operation.

1. Log on to the server that contains the Microsoft AD schema (typically the schema is on the domain controller).
2. Launch a command prompt, using the Run as Administrator option.
3. Execute the following command
   
   regsvr32 schmmgmt.dll
4. Click the **Start** button and enter **mmc.exe** in the search field.

5. Launch the **mmc.exe** program.

6. Click on the **File** menu. Then, click **Add / Remove Snap-in**.

7. Click on **Active Directory Schema** in the list of **Available snap-ins** on the left; then, click **Add** to add it to the **Selected snap-ins on the right**.

8. Click **OK**.
9. Expand the **Active Directory Schema** on the left.

**6.2.1.2 Reviewing Existing Attributes to Avoid Redundancies when Creating New Attributes**

Before you create a new attribute it is important to review existing user attributes in your Active Directory Schema. Under Active Directory Schema on the left, expand the Classes folder and scroll down to click on the **user** class. Examine the existing set of **user** class attributes listed on the right. These attributes are native to Active Directory, and can be assigned to users as subject attributes. These attributes may meet existing requirement for implementing subject attribute, alleviating the need to add custom attributes to the schema. You can list the attributes in alphabetic order by clicking on the **Name** column.
Let’s say you wanted to create an attribute to store the user’s cell phone number, you would look through the attributes and notice that the attribute **cellphone** does not exist. However, there is an existing attribute named **mobile** that could be used to store a cell phone number.

Once you have identified that the creation of a new attribute is warranted, proceed with the instructions in the following section.
6.2.1.3 Creating New Custom Attributes

1. Launch a browser window and go to the Microsoft site:
   https://gallery.technet.microsoft.com/scriptcenter/56b78004-40d0-41cf-b95e-6e795b2e8a06

2. Copy the oidgen.vbs script code that is shown on the page to the clipboard.

3. Open Notepad and paste the script into the editor.

4. Save the script to a file on the desktop named oidgen.vbs.

5. Go back to the Active Directory schema window.

6. On the left pane and click on the Attributes folder.

7. Right click on the Attributes folder and select Create Attribute.

8. Click Continue on the warning window.

9. Enter the name of your new attribute and select the type of attribute in the Syntax field. In the example below, the name of the new attribute is clearance and the type of attribute is Unicode String.
6.2.1.4 Generating an ID to Enter into the Unique X500 Object ID Field

Next you need to generate an ID to enter into the Unique X500 Object ID field.

1. Go to the desktop and double click on the `oidgen.vbs` script that was saved earlier. This should execute the script to generate a unique Object ID.

2. Enter this long Object ID into the **Unique X500 Object ID** field in the Active Directory Create New Attribute window.

3. Click **OK** to create the new attribute.

4. Scroll down the list of attributes and make sure your newly added attribute is listed there.
6.2.1.5 Adding the New Attribute to the User Class

Next you need to add the new attribute to the user class.

1. In the left pane, expand the Classes folder. Scroll down the list of classes and right click on the user class and select Properties.

2. Click on the Attributes tab.

3. Click Add. Scroll down and click on the new attribute.
4. Click **OK** on the Select Schema Object window, and then click **OK** one more time on the User Properties window. At this point you’ve added the new attribute to the user class.

When you examine the list of attributes for the **user** class you should be able to see the new attribute.

### 6.2.2 Set Values for Custom User Attributes in Microsoft AD

Once you've created a new custom attribute in the Active Directory **user** class, that new attribute will be available for all users in the domain. You will be able to set specific values for the new attribute for each distinct user. Follow the instructions in this section to set a user-specific value for a new attribute in Active Directory.

1. Log on to the Microsoft AD server.
2. Open the Active Directory Users and Computers program.
3. Click on the **View** menu and select **Advanced Features**.

4. Right click on **Saved Queries** and select **New > Query**. Enter a name for your query (e.g. **My Users**).

5. Click on **Define Query**. From the **Name** list, select **Has a value**.
6. Click OK. Then, click OK again to create your new query.
7. You will see a list of Active Directory Users displayed in the right pane.
8. Double click on the specific user (e.g. Lucy Smith) that you want to modify to bring up the properties window.
9. Click on the **Attribute Editor** tab.

10. Scroll down and locate the new custom attribute you want to set a value for (e.g. **clearance**).
11. Double click on the attribute, and enter a value suitable for your organization. In this example the clearance attribute will be set to a value of **Interim** for the user **Lucy Smith** in subsequent steps.

12. Click **OK** and then click **OK** again. The information is saved and the User Properties window closes.

![Image of User Properties window](image)

**Note:** When you set an attribute value in the attribute editor and then go back to the Users query view, you have to press F5 or click the **Action menu > Refresh** to see the new value in the view.

### 6.2.2.1 Adding New Columns to the Users Query View

Next you will add new columns to the Users query view to help monitor the custom attribute values for each user in the directory. By default, the Users view only shows the attribute values for **Name**, **Type** and **Description**.
1. In the Saved Queries folder, click on the name of the query to be modified (e.g. My Users).
2. Click on the View menu and select Add/Remove Columns...
3. In the list of Available columns, scroll up or down to find desired columns.
4. Click on column name and click on the Add button.
5. When all desired columns have been chosen click OK.

The following screenshot shows a query view after adding custom attribute columns. The example contains new columns for the attributes User Logon Name, Company, Department, Title, Staff Level, and Clearance.
6.3 Configure PingFederate Servers to Pull User Attributes

6.3.1 Configure PingFederate-IdP to Pull User Attributes During Authentication

Follow the instructions in this section to configure the PingFederate-IdP to pull user attribute values from Microsoft AD during the authentication process. In the following example, the value for the user attribute company is extracted from Microsoft AD.

1. Launch your browser and go to: https://<DNS_NAME>:9999/pingfederate/app.
2. Replace DNS_NAME with the fully qualified name of the Identity Provider’s PingFederate server (e.g. https://idp.abac.test:9999/pingfederate/app).
3. Log on to the PingFederate application using the credentials you configured during installation.
4. On the Main menu under SP CONNECTION, click Manage All SP.

5. Click on the link for the connection created in chapter 3 (e.g. https://rp.abac.test:9031).
6. On the Activation & Summary screen, scroll down to the **Assertion Creation** group and click on the **ATTRIBUTE CONTRACT** link.

7. On the Attribute Contract screen, under the **EXTEND THE CONTRACT** column, enter the name of the attribute to be extracted from Microsoft AD (e.g. *company*) in the empty text field.
8. Click **Add**.

9. Click **Next**.

10. On the Authentication Source Mapping screen click on the name of the **ADAPTER INSTANCE** that is listed (e.g. **RSA Multifactor**).
11. Click on **Assertion Mapping** tab and select **Retrieve additional attributes from multiple data stores using one mapping.**

12. Click **Next.**

13. Click on **Add Attribute Source.**
14. On the Attribute Sources & User Lookup screen enter a unique name in the **Attribute Source Id** field (e.g. ActiveDirectory).

15. In the **Attribute Source Description** field, enter a description.

16. From the **Active Data Store** list, select the existing Data Store that connects to Active Directory.

17. Click **Next**.

18. On the LDAP Directory Search screen, enter the **Base DN** (e.g. DC=ABAC,DC=TEST).

19. Under the **ROOT OBJECT CLASS** column, select the Active Directory class that contains the attribute you want to pull the value from. In the example below, the **organizationalPerson** class is selected because it is the root class that contains the company attribute.

20. Under the **ATTRIBUTE** column, select the attribute (e.g. **company**), then click **Add Attribute**.
21. Click **Next**.

22. On the LDAP Filter screen, enter `samaccountname=${username}`.

23. Click **Next**.
Chapter 6. Attribute Exchange Between the Identity Provider and Relying Party

24. On the Summary screen, click **Done**.

25. On the Attribute Sources & User Lookup screen, click **Done**.
26. On the Attribute Contract Fulfillment screen, for the company attribute select the **SOURCE** and **VALUE**. For the **SOURCE**, select LDAP (Atts from MS AD). For **VALUE** select company.

27. Click **Save** to complete the configuration.

### 6.3.1.1 Functional Test of Pulling User Attributes During Authentication

The instructions in this section will help perform a test to ensure that the Identity Provider is getting the configured attributes (e.g. **company**) from Active Directory and passing them in a SAML message to the Relying Party. The Firefox SAML tracer Add-on is used to examine the SAML message.

Follow the instructions in section 6.6.1, Temporarily Disable SAML Encryption for Testing and Troubleshooting Message Exchanges, on page 240 to disable SAML encryption. Once SAML encryption has been disabled, you can proceed with the following functional test instructions.

1. Launch your Firebox browser and select **SAML tracer** from the **Tools** menu.
   
   This launches an empty SAML tracer window.

2. Minimize the SAML tracer window.

   The SAML tracer automatically records the details of the HTTPS messages in the background.
3. Go back to the main browser window and go to the Relying Party’s SharePoint site (e.g. https://SharePoint.abac.test).

4. Select **Federated Logon from Identity Provider**.

5. In the Identity Provider’s PingFederate Sign On screen, enter the credentials for the account you are testing with (e.g. Ismith) and click **Sign On**.

6. On the RSA 2-factor authentication screen, enter the validation code and proceed.

   The browser redirects to the PingFederate-RP and then to the Relying Party’s SharePoint site. You may not notice the redirection to the PingFederate-RP if it happens quickly.

7. Go back to the SAML tracer window. Scroll down and click on the last **POST** message that contains a SAML icon.

8. Click on the **SAML** tab. Scroll down the SAML message and locate the **AttributeStatement** node and sub nodes.
Expected Result: Ensure that the attribute you configured from Microsoft AD contains a node. In the preceding example screen shot you can see that there is an Attribute node for the company attribute because of the line `<saml:AttributeValue>`

Expected Result: Ensure that the AttributeValue node contains the expected value for the attribute from ActiveDirectory. In the example screen shot above you can see there is an AttributeValue node for the company attribute and the value is Conway Inc. This is correct because in our Microsoft AD environment, the user account we tested with is Ismith (Lucy Smith), and Lucy’s company attribute in Microsoft AD is set to a value of Conway Inc.

When you complete this functional test, you must enable SAML encryption between the Identity Provider and Relying Party again. Follow the instructions in the section 6.6.1.2, Enable SAML Encryption Again, on page 241 to enable SAML encryption.

6.3.2 Configure PingFederate-IdP to Pull Environmental Attributes During Authentication

Follow the instructions in this section to configure the PingFederate-IdP to get environmental attribute values from the RSA Adaptive Authentication system during the authentication process. The environmental attributes are passed along with the user attributes in the SAML messages that is sent to the Relying Party. In the example below, the environmental attribute ip_address will be pulled from RSA Adaptive Authentication.

RSA Adaptive Authentication stores environmental attributes about the user's web transactions in a SQL Server database named RSA_CORE_AA. The PingFederate-IdP will be configured to query to the RSA_CORE_AA database and get the value of ip_address from the EVENT_LOG table.

Before you can configure the query for ip_address, you must first create an account for the PingFederate application in the RSA_CORE_AA database. Follow these instructions to create the account in the SQL Server database.
Log on to the server that hosts the RSA Adaptive Authentication SQL Server database engine.

1. Open SQL Server Management Studio.
2. Expand the RSA-AA-Server folder, then the Security folder.
3. Right click on Logins and select New Login.

4. Set the Login name (e.g. ping), under SQL Server authentication choose a password that meets the Windows password policy.
5. Under Server Roles, select public.

6. Under User Mapping, check the Map box next to RSA_CORE_AA. In the bottom pane, under Database role membership, check the box next to db_datareader.

7. Under Status, set Permission to connect to database engine to Grant and Login to Enabled. Click OK.
6.3.2.1 Configuring a New Data Store that Connects to the RSA Database

Next you will configure a new Data Store that connects to the RSA_CORE_AA database on the Identity Provider’s PingFederate server. This new data store will be used in the RP Connection to query the EVENT_LOG table during the authentication process.

Follow the instructions below to create a new Data Store for the RSA_CORE_AA database.

1. Launch your browser and go to: https://<DNS_NAME>:9999/pingfederate/app. Replace <DNS_NAME> with the fully qualified name of the Identity Provider’s PingFederate server (e.g. https://idp.abac.test:9999/pingfederate/app).

2. Log on to the PingFederate application using the credentials you configured during installation.

3. Under Server configuration, select Data Stores.
4. Under **Manage data stores**, select **Add new data store**. Select **Database** as type of data store. Click **Next**.

5. On the database config page, set the JDBC URL to:
   
   `jdbc:sqlserver://<RSA_SERVER_IP_ADDRESS>:1433;databaseName=RSA_CORE_AA`

   Replace `<RSA_SERVER_IP_ADDRESS>` with the IP address of the server that hosts the RSA_CORE_AA database.

6. Set the driver class to `com.microsoft.sqlserver.jdbc.SQLServerDriver`.

7. In the **Username** and **Password** fields, enter the credentials for the ping user created in the SQL server RSA database.

8. Under **Validate Connection SQL**, type **SELECT 1=1**.

9. Select the check box **Allow multi-value attributes**; then, click **Next**.

10. Review the settings on the summary page. Then, click **Save**.
6.3.2.2 Modifying the SP Connection to the RP to Add New Environmental Attribute

Next you will modify the SP Connection to the Relying Party and add a new environmental attribute ip_address from the RSA_CORE_AA database.

1. Go to the PingFederate Main menu.
2. On the Main menu under SP CONNECTION, click Manage All SP.
3. Click on the link for the SP connection created in chapter 2 (e.g. https://rp.abac.test:9031).
4. On the Activation & Summary screen, scroll down to the **Assertion Creation** group and click on the **ATTRIBUTE CONTRACT** link.

5. On the Attribute Contract screen, under the **EXTEND THE CONTRACT** column, enter the name of the environmental attribute to be pulled from the **RSA_CORE_AA** database (e.g. **ip_address**) in the empty text field.

6. Click **Add**.
7. Click **Next**.

8. On the Authentication Source Mapping screen click on the name of the **ADAPTER INSTANCE** (e.g. **RSA Multifactor**).
9. Click on the Attribute Sources and User Lookup tab.

10. Click Add Attribute Source.

11. On the Attribute Sources & User Lookup screen, enter a unique name in the Attribute Source Id field (e.g. RSAEventLog).

12. Enter a description (e.g. Atts from RSA).

13. For the Active Data Store field, select the existing Data Store that connects to the RSA_CORE_AA database.

14. Click Next.

15. On the Database Table and Columns screen, select the dbo Schema.

16. Select the EVENT_LOG table.

17. Under the Columns to return from SELECT, select the IP_ADDRESS column and click Add Attribute.
18. Click **Next**.

19. On the Database Filter screen, enter the text on the following line into the text field for the **Where**. Make sure to include the quotes.

\[
\text{EVENT_ID} = '{\text{transactionid}}'
\]

20. Click **Next**.
21. On the Summary screen, click **Done**.

22. On the Attribute Sources & User Lookup screen, click **Done**.
23. On the **Attribute Contract Fulfillment** screen, for the **ip_address** attribute select the **SOURCE** and **VALUE**. For the **SOURCE**, select **JDBC (Atts from RSA)**. For **VALUE** select **IP_ADDRESS**.

24. Click **Save** to complete the configuration.

### 6.3.2.3 Functional Test of Pulling Environmental Attributes During Authentication

To test that the Identity Provider’s PingFederate server is successfully getting the environmental attributes during the authentication process, follow the instructions in section 6.3.1.1, **Functional Test of Pulling User Attributes During Authentication**. The only exception to those instructions is that when you examine the SAML message, you need to look for the environmental attribute that is being pulled from the **RSA_CORE_AA** database. See below for an example.
1. Once you have the message open in the SAML tracer window, scroll down the message and locate the **AttributeStatement** node and sub-nodes.

   ![SAML Message](image)

   **Expected Result:** Ensure that the attribute you configured to be pulled from the RSA_CORE_AA database contains a node. In the preceding example screen shot you can see that there is an Attribute node for the **ip_address** attribute because of the line `<saml:Attribute Name="ip_address" ...>

   ![SAML Attribute](image)

   **Expected Result:** Ensure that the AttributeValue node contains the expected value for the attribute from the RSA_CORE_AA database. In the preceding example screen shot you can see there is an AttributeValue node for the **ip_address** attribute and the value is `10.255.207.19`.

6.3.3 Configure PingFederate-RP to Pull Attributes from the Identity Provider's SAML Exchange

Once the PingFederate-IdP completes the authentication for a user, the Identity Provider will send a SAML message to the PingFederate-RP. That SAML message will contain attributes.

Follow the instructions below to configure the PingFederate-RP to get attributes and their associated values from the SAML message exchange with the Identity Provider. In the example below, the attribute being configured at the Relying Party is the **company** attribute.

1. Launch your browser and go to: https://<DNS_NAME>:9999/pingfederate/app. Replace `DNS_NAME` with the fully qualified name of the Relying Party’s PingFederate server (e.g. https://rp.abac.test:9999/pingfederate/app). Log on to the PingFederate application using the credentials you configured during installation.

2. On the **Main** menu, under **IDP CONNECTIONS**, click on the connection that was configured to the Identity Provider in chapter 3 (e.g. https://idp.abac.test:9031).
3. On the Activation & Summary screen, scroll down to the User-Session Creation group and click on the ATTRIBUTE CONTRACT link.

4. On the Attribute Contract screen, under the EXTEND THE CONTRACT column, enter the name of the attribute to be pulled from the Identity Provider’s message (e.g. company) in the empty text field. In the ACTION column, click Add.
5. Click **Done**.

6. On the User-Session Creation screen, click **Configure User-Session Creation**.
7. On the Summary page, under **User-Session Creation**, click on the **CONNECTION MAPPING CONTRACT** link.

8. On the Connection Mapping Contract screen, make note of the **CONNECTION MAPPING CONTRACT** being used because you will need to modify it by adding new attributes. In the example screen shots the contract name is **SharePoint 2013**.

9. Click on **Manage Connection Mapping Contracts**.

10. On the Manage Contracts screen, click on the name of the contract that is being used for the current configuration (e.g. **SharePoint 2013**).

12. On the Contract attributes screen, under the EXTEND THE CONTRACT column, enter the name of the attribute to be shared with the PingFederate service provider connection (e.g. company).

13. In the ACTION column, click Add.

14. Click Done.

15. On the Manage Contracts screen, click Save.

On the Connection Mapping Contract screen you should see the new attribute (e.g. company) listed on the page.
16. Click on the **Contract Fulfillment** tab.

17. On the Contract Fulfillment screen, for the new attribute (e.g. company) select **Assertion** for the **SOURCE** field and select **company** for the **VALUE** field.

18. Click **Save** to complete the configuration.
6.4 Configure PingFederate-RP and SharePoint to Pass and Read Attributes

6.4.1 Configure PingFederate-RP to Pass Attributes to SharePoint

Once the PingFederate-IdP completes the authentication for a user, the Identity Provider will send a SAML message to the PingFederate-RP. That SAML message will contain attributes. The PingFederate-RP will then take the attributes and send them to SharePoint via WS-Federation.

Follow the instructions below to configure the PingFederate-RP to pass attributes and their associated values from the Identity Provider to SharePoint. In the example below, the attribute being configured to be passed to SharePoint is the company attribute.

1. Launch your browser and go to: https://<DNS_NAME>:9999/pingfederate/app. Replace DNS_NAME with the fully qualified name of the Relying Party's PingFederate server (e.g. https://rp.abac.test:9999/pingfederate/app).

2. Log on to the PingFederate application using the credentials you configured during installation.

3. On the Main menu under SP CONNECTION, click Manage All SP.

4. Click on the link for the WS-Federation connection to the SharePoint instance created in chapter 3 (e.g. SharePoint).

5. On the Activation & Summary screen, scroll down to the Assertion Creation group.

6. Click on the ATTRIBUTE CONTRACT link. On the Attribute Contract screen, under the EXTEND THE CONTRACT column, enter the name of the attribute (e.g. "company") to be
passed from the PingFederate-RP to SharePoint in the empty text field. For the ATTRIBUTE NAME FORMAT select the schemas.xmlsoap.org 2005 identity claims format.

7. Click Add.

8. Click Done.
9. On the Authentication Source Mapping screen, under the CONNECTION MAPPING CONTRACT NAME heading click on the name of the connection mapping contract (e.g. SharePoint 2013) between this PingFederate SP connection and the PingFederate IdP connection that was configured in section 6.3.3, Configure PingFederate-RP to Pull Attributes from the Identity Provider’s SAML Exchange.

10. On the Attribute Contract Fulfillment screen, for the company attribute, select Connection Mapping Contract for the SOURCE field. Select company for the VALUE field.

11. Click Save to complete the configuration.

6.4.1.1 Functional Test of PingFederate-RP Passing Attributes to SharePoint

The instructions in this section will help perform a test to ensure that the PingFederate-RP is sending the correct attributes to SharePoint. The Firefox SAML tracer Add-on is used to examine the SAML message.
1. Launch your Firebox browser and select **SAML tracer** from the **Tools** menu.
   This will launch an empty SAML tracer window. Minimize the SAML tracer window. The SAML tracer will automatically record the details of the HTTPS messages in the background.

2. Go back to the main browser window and go to the Relying Party’s SharePoint site (e.g. https://SharePoint.abac.test).

3. Select the option to use the federated logon (e.g. **Federated Logon from Identity Provider**).
   Your browser should be redirected to the PingFederate-IdP and you should see the PingFederate Sign On screen.

4. Enter the **Username** and **Password** of the Microsoft AD account created previously in this guide (e.g. Ismith).
5. Click **Sign On**. On the RSA Adaptive Authentication screen, enter the SMS validation code received on your mobile phone. Click **Continue**.

Once authenticated at the Identity Provider, your browser should automatically redirect to the PingFederate-RP (e.g. `rp.abac.test`) and then to the Relying Party’s SharePoint (SharePoint.abac.test) site.

6. Go back to the SAML tracer window. Scroll down the list of messages and click on the **POST** message to **SharePoint_trust** URL to bring up the details of the message in the bottom pane.

7. Click on the **Parameters** tab for the bottom pane.
8. Copy all of the content (beginning with the POST line) in the bottom page and paste it into a text editor such as Notepad. Turn on Word Wrap to make it easier to see all of the XML content.

9. Scroll down the SAML message and locate the AttributeStatement node and sub-nodes.

10. For the AttributeStatement node and sub-nodes, enter some carriage returns before each XML tag to make it easier to examine the data. The goal is to be able to easily examine the Attribute nodes within the AttributeStatement node.
Expected Result: Within the AttributeStatement node, there should be multiple Attribute sub-nodes. There should be an Attribute sub-node that has an AttributeName value of company. The AttributeNamespace value should be http://schemas.xmlsoap.org/ws/2005/05/identity/claims. There should be an AttributeValue sub-node and it should contain the expected value (e.g. Conway Inc) for the company attribute that was pulled from Microsoft AD (e.g. <saml:AttributeValue>Conway+Inc</saml:AttributeValue>) for the specific user (e.g. Ismith) that authenticated at the Sign On screen.

6.4.2 Configure SharePoint to Read Custom Attributes from PingFederate-RP

The PingFederate-RP will send attributes to SharePoint via WS-Federation. Follow the instructions below to configure SharePoint to read the attributes and load them into the web session. In the example below, the attribute being configured to be read by SharePoint is the company attribute.

1. Using SharePoint administrator credentials, log on to the server that hosts SharePoint for the Relying Party.

2. Click on the Start menu and navigate to SharePoint 2013 Products group. Open SharePoint 2013 Management Shell.
3. Enter each of the commands displayed below the next paragraph into the management shell to configure a new attribute, company for the existing Trusted Identity Token Issuer named Federated Logon from Identity Provider. Enter each command separately, and enter a carriage return after the command. If the command executed successfully, management shell will not provide any feedback. If an error occurs, the management shell will display the error.

```
$tokenIssuer = Get-SPTrustedIdentityTokenIssuer -Identity "Federated Logon from Identity Provider"
$tokenIssuer.Update()
$claimmap = New-SPClaimTypeMapping -IncomingClaimType "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/company" -IncomingClaimTypeDisplayName "company" -SameAsIncoming
```


```
$tokenIssuer = Get-SPTrustedIdentityTokenIssuer -Identity "Federated Logon from Identity Provider"
$tokenIssuer.Update()
$claimmap = New-SPClaimTypeMapping -IncomingClaimType "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/company" -IncomingClaimTypeDisplayName "company" -SameAsIncoming
Add-SPClaimTypeMapping -TrustedIdentityTokenIssuer $tokenIssuer -Identity $claimmap
```
6.4.2.1 Functional Test of SharePoint Reading Attributes from PingFederate-RP

The instructions in this section will help perform a test to ensure that SharePoint can read the attributes sent in messages from the PingFederate-RP.

1. Follow the instructions in this section to ensure that SharePoint is configured to read the newly configured attributes from PingFederate-RP.

2. Launch your browser and go to the SharePoint central administration page (e.g. http://SharePoint.abac.test:44444/default.aspx).

3. Log on using the credentials of the SharePoint administrator.

4. Under the Application Management group, click on Manage Web Applications.

5. Click on the web application that contains the SharePoint site you are managing (e.g. SharePoint - 80). SharePoint highlights the web application row that you clicked.
6. Click User Policy.

7. Click the **Add users** link.
8. Click **Next**.

9. On the **Add Users** screen, click the small browse icon (looks like an open book) under the **Users** field.

**Expected Result:** On the Select People and Groups screen, you should see a grouping with the name of the trusted token issuer (e.g. **Federated Logon from Identity Provider**). You should also see the newly configured attribute (e.g. **company**) listed under that grouping.
6.5 Configure the Claims Viewer Web Part at the SharePoint Site

Follow the instructions in this section to configure the Claims Viewer Web part at the SharePoint site. The Claims Viewer is a component that is useful to the SharePoint administrator because it displays a list of the attributes that are loaded into the web session. This list can be used to validate that the correct set of attributes and associated values are being passed from the PingFederate-RP, and that SharePoint is correctly configured to read the attributes.

1. Log on to the server that hosts SharePoint for the Relying Party.

2. Launch your browser and go the SharePoint central administration page (e.g. http://SharePoint.abac.test:44444/default.aspx). Log on using the credentials of the SharePoint administrator.

The central administration home page displays.

3. On the Central Administration menu on the left, click System Settings.

4. On the Farm Management menu, click Manage Farm Solutions.
5. Click on the `helloitsliam.claimsviewerwebpart.wsp` link.

6. Click on the **Deploy Solution** link at the top of the page.

7. Click **OK** at the bottom of the page.

   The claimsviewerwebpart should be shown as deployed on the Solution Management page.
Chapter 6. Attribute Exchange Between the Identity Provider and Relying Party

6.5.1 Configure SharePoint Claims Viewer

This section explains how to add a new page to the SharePoint site to view the claims.

1. Log on to the Relying Party's SharePoint site (e.g. https://SharePoint.abac.test) using the credentials of the SharePoint administrator. Select Windows Authentication on the Sign On screen.

2. Click the gear icon at the top right corner of the page and select the Site Contents link.
3. Click on the Site Pages library. This will show a list of the existing pages on the site.

4. Click the **new Wiki page** link to add a new page. This link may be named differently, depending on the type of SharePoint template your site is configured with. Enter a name for the new page (e.g. **ClaimsView**).
5. Click **Create**. The SharePoint page editor for the newly added page displays.

6. Click on the **INSERT** tab at the top of the page. Click on the **Web Part** button.

7. From the **Categories** list, select **Custom**. From the **Parts** list, select **ClaimsViewerWebPart**.
8. Click Add.

9. Click the SAVE button at the top right corner of the page.

SharePoint launches the new page (e.g. ClaimsView) that was just created. (Save the URL of the new page (e.g. https://SharePoint.abac.test/SitePages/ClaimsView.aspx), because you will use it later in a functional test.)

The Claims Viewer Web Part on the page displays. It is collapsed by default.
10. Click on the + sign under **ClaimsViewerWebPart** to view the claims data. You see a list of claim values, and information about the SAML token at the bottom of the page.

### 6.6 Functional Test of All Configurations for this Chapter

The instructions in this section will perform an integrated test all of the configurations in this chapter. Using the browser, you will log on using an account that was created in Microsoft AD. Then you will use the SharePoint claims viewer to validate that the newly configured attributes are passed from the Identity Provider to the Relying Party and that the attributes are successfully loaded into the SharePoint web session.
1. Launch your browser and go to the Relying Party’s SharePoint site (e.g. https://SharePoint.abac.test).

2. Select **Federated Logon from Identity Provider**.
   Your browser is redirected to the PingFederate-IdP and you see the PingFederate Sign On screen.

3. Enter the credentials of the Microsoft AD account created earlier in this guide (e.g. Ismith).
4. Click **Sign On**. On the RSA Adaptive Authentication screen, enter the SMS validation code received on your mobile phone. Then, click **Continue**.

Once authenticated at the Identity Provider, your browser automatically redirects to the PingFederate-RP (e.g. rp.abac.test) and then to the Relying Party's SharePoint site (SharePoint.abac.test).

5. Once you arrive at the SharePoint site home page, navigate to the claims viewer page that was created in the previous section (e.g. https://SharePoint.abac.test/SitePages/ClaimsView.aspx). Expand the claims viewer web part on the page to see a list of claims.

**Expected Result:** You should see the newly configured attribute (e.g. `company`), and its associated claim value. The claims viewer shows the name of each attribute (i.e. `claim`) using a long format such as

6.6.1 Temporarily Disable SAML Encryption for Testing and Troubleshooting Message Exchanges

Follow the instructions below to temporarily disable the encryption of SAML messages between the Identity Provider and the Relying Party. You should only perform the steps in this section when explicitly instructed to do so in another section of the guide (e.g. during a functional test). You may also need to refer back to this section in the future to test or troubleshoot SAML message exchanges in your environment.

Temporarily disabling the encryption can help test that the expected attributes are being exchanged between the Identity Provider and the Relying Party. By temporarily disabling the encryption, you will be able to see the attributes and their associated values in the SAML messages using the Firefox SAML tracer Add-on or a comparable software tool. When testing or troubleshooting has completed, you can enable the encryption again.

6.6.1.1 Disable SAML Encryption

1. Launch your browser and go to: http://<DNS_NAME>:9999/pingfederate/app. Replace DNS_NAME with the fully qualified name of the Identity Provider’s PingFederate server (e.g. https://idp.abac.test:9999/pingfederate/app). Log on to the PingFederate application using the credentials you configured during installation.

2. On the Main menu under SP CONNECTION, click Manage All SP.

3. Click on the link for the SP connection that you want to disable the encryption for (e.g. https://rp.abac.test:9031).

4. Scroll down to the Protocol Settings group.
5. Click on the **ENCRYPTION POLICY** link.

6. On the Encryption Policy screen, select **None**.

7. Click **Save**.

At this point you have disabled SAML encryption at the Identity Provider for this specific connection to the Relying Party. You can perform authentication testing using the Firefox SAML tracer to examine the SAML messages being sent by the Identity Provider to the Relying Party.

### 6.6.1.2 Enable SAML Encryption Again

Once testing has completed, perform the following instructions to enable the encryption once again.

1. On the PingFederate **Main** menu under **SP CONNECTION**, click **Manage All SP**.

2. Click on the link for the SP connection that you want to enable the encryption for (e.g. `https://rp.abac.test:9031`).

3. Scroll down to the Protocol Settings group.
4. Click on the **ENCRYPTION POLICY** link.

5. On the Encryption Policy screen, select **The entire assertion**.

6. Click **Save**.

7. On the Select XML Encryption Certificate screen, select the **Block Encryption Algorithm** (e.g. **AES-128**) and the **Key Transport Algorithm** (e.g. **RSA-OAEP**). For the selection box above **Manage Certificates**, select the Relying Party’s public key certificate to be used to encrypt the message content.
8. Click **Save**.

You have now enabled the encryption for the connection again.
# Setting up NextLabs to Protect SharePoint

## 7.1 Introduction

## 7.2 Components

## 7.3 Installation and Configuration of NextLabs Control Center (on the SQL Server)


## 7.5 Installation and Configuration of Policy Controller (PDP)

## 7.6 Installation and Configuration of NextLabs Entitlement Manager for SharePoint Server

## 7.7 Functional Tests
7.1 Introduction

In this build we are using an ABAC architecture to protect resources on a Microsoft SharePoint instance. In this section we will install the NextLabs Control Center, Policy Studio, Policy Controller, and Entitlement Manager for SharePoint Server. Before getting started installing these components, you must prepare your environment. At a minimum, Windows Server 2012 must be set up with a configured Active Directory, and SharePoint must be installed and configured with a Site Collection. If you haven’t already completed the basic installation and configuration of Windows Server 2012 and Active Directory, please refer to chapter 2, Setting up the Identity Provider. If you have not already completed the installation and configuration of SharePoint, please refer to chapter 3, Setting up Federated Authentication Between the Relying Party and the Identity Provider.

The four NextLabs components installed in this chapter provide an Information Control Platform (ICP), Policy Administration Point (PAP), Policy Decision Point (PDP), and Policy Enforcement Point (PEP) in the ABAC Architecture. Each component will be described generally in section 7.2, Components. Then there will be separate sections illustrating installation and configuration of each component. Finally, section 7.7, Functional Tests, will give some guidance for verifying the correct installation and configuration of the various components presented in this chapter.

7.2 Components

- **NextLabs Control Center (release 7.5):** enterprise-level Information Control Platform (ICP) for policy-driven data loss prevention and entitlement management; can contain many software components, including the following two in this build:
  - **Policy Studio: Enterprise Edition (PAP):** application for policy lifecycle management, provides a graphical user interface (GUI) for defining and deploying attribute-based access control policies. This product is installed on an instance of SQL Server.
  - **Policy Controller (PDP):** distributed component of the Control Center that evaluates policies created in the PAP to determine a deny or allow decision when users attempt to access protected resources. This product is installed on an instance of Microsoft SharePoint Server.

- **NextLabs Entitlement Manager for Microsoft SharePoint Server (PEP):** enforces the decisions from the PDP to deny or allow access to SharePoint resources. This product is installed on an instance of Microsoft SharePoint Server.

7.2.1 NextLabs Control Center (release 7.5)

The NextLabs Control Center is an enterprise-level Information Control Platform (ICP). It integrates into existing IT infrastructure, and applications and can be used to digitally manage policies to govern data classification, access, sharing, and automate security compliance procedures. In order to fulfill its diverse capabilities, the Control Center can be configured to incorporate and coordinate many NextLabs software components. It is also possible to develop your own custom access control enforcers for applications that do not already have an available enforcer built by NextLabs. In this build, we take advantage of the Policy Studio, Policy
Controller, and Entitlement Manager for Microsoft SharePoint Server, which are discussed in the following sub-sections.

In order to support administrative and configuration activities necessary for its many components, NextLabs Control Center provides a web application user interface called Administrator. Some of the system monitoring and administrative tasks available via Administrator include: checking how many policies are deployed in the network, finding out on which hosts the Control Center components are installed, checking the status of Control Center server components, finding out how many enforcers are currently running, finding out if any enforcers are disconnected, and finding out or modifying the current heartbeat setting for an enforcer, among others.

Another key component of the Control Center is the Policy Server. The Policy Server runs continuously from the moment of startup as a Windows service. As new policy is defined or policies are updated, the Policy Server pushes these policy sets to the Policy Controller on the SharePoint Server.

The Control Center platform is installed and configured on the same server as the build's SQL database, which we refer to as the SQL Server.

### 7.2.2 NextLabs Policy Studio: Enterprise Edition

The NextLabs Policy Studio component of the Control Center is intended for administrators and policy designers responsible for converting the general data access and usage management goals of the enterprise into deployable, active policies. Depending on a company’s business rules, policies can be defined to evaluate user (subject) attributes, resource (object) attributes, and environmental (contextual) attributes.

The Policy Studio provides a graphical user interface with which you can create an abstract model representing the various parts of the enterprise environment (users, applications, computers, and environmental context), construct policies with these modeled components, and fine-tune policies using advanced conditions that can change based on dynamic comparisons, evaluations, and contextual factors. For example, policy designers can select pre-defined conditions including the time of day, day of the week, connection type, and IP address, among many others. In addition to defining which attributes to evaluate when making an enforcement decision, the policy construction process can also determine notification obligations such that when a policy is allowed or denied, a user can be notified with a default or custom message, a statement can be added to the application’s log file, and an email can be sent to an administrator.

Like the Control Center platform, the Policy Studio is installed and configured on the SQL Server.

### 7.2.3 NextLabs Policy Controller

Each NextLabs Policy Controller provides the interface to the Policy Server component of the Control Center (installed on the SQL Server), and serves as a distributed Policy Decision Point (PDP). It comprises a set of software modules delivered with Control Center, read-to-install on the enforcer host or development machine. Because it is not specific to any adapter type, it requires no customization. In this build, the Policy Controller is installed and configured on the same server as the SharePoint instance, which we refer to as the SharePoint Server.
In general, the logical architecture of a NextLabs enforcer that protects an application (such as the Entitlement Manager for SharePoint Server, covered in the next sub-section) consists of two parts, the Policy Controller and the Policy Adapter.

The Policy Controller consists of the following functional components:

- The **Policy Evaluation Engine** evaluates whether or not each user action is covered by any of the policies currently cached at that enforcement point. It bases its evaluation on multiple criteria such as who the user is, what host he is using, how he is connected to the network, which action is being attempted, on what resource, the date, the time, and so on. It does this in real time, and operates continuously whether the host is connected to the network or not. Note that while disconnected from the network the local encrypted bundle.bin policy cache would not be able to be updated from policy changes made in the PAP.

  - Note: Policies are authored in the PAP GUI on the SQL Server, and any modifications to the policy set are transmitted by the Policy Server, also installed on the SQL Server, to the Policy Controller on the SharePoint Server. It takes a heartbeat length of time for the updates to take effect on the SharePoint Server. By default, the heartbeat rate of the desktop enforcer is set to 60 minutes, which is appropriate for a live production environment. For testing and learning purposes, however, you should change this to 1 minute, which will allow you to define, deploy and test policies with shorter delays. A heartbeat can be configured via the Control Center Administrator web application.

- The **Context Manager** keeps constant track of the environmental context of all events, and provides it to the Policy Engine and Policy Adapter. The context includes user identity, computer host name, network connection type, and date and time.

- For any policy that evaluates as True, the **Obligation Manager** initiates an obligation by sending a request to a policy adapter's obligation services or executing a built-in obligations. It contains three sub-components:

  - **Policy Logger** - collects and logs all activity details and policy decision results
  - **Messaging Services** - sends message to recipients or targets listed in a policy
  - **Application Extender** - launches an application or custom executable that performs some custom obligation

- The **Controller Manager** records non-policy activities, updates the configuration, and secures the controller. Components include:

  - **Activity Recorder** - records activities tracked by the policy adapter in real time.
  - **Configuration Manager** - applies profile and system configuration changes in real time
  - **Policy Authentication** - authenticates the policy set from the Policy Server and encrypts it on the local file system

  - Note: It is the responsibility of the Controller Manager to encrypt the bundle.bin file on the local file system for use during policy evaluation by the PDP.

  - **Tamper Resistance Module** - protects all Entitlement Manager processes, installed files, and registry settings from tampering by users or other processes, and governs the automatic start-up and restart features. The Policy Controller runs as a Windows service continuously from the moment of startup, called Control Center Enforcer Service.
The ICENet Client provides the interface for all communication with the Policy Server. It is used for deploying new or changed policies, periodically sending activity logs from each control point, and providing controller health status.

### 7.2.4 NextLabs Entitlement Manager for Microsoft SharePoint Server

The NextLabs Entitlement Manager for SharePoint is designed to enforce the policies that control whether and how users can access, download, and use data stored on a SharePoint server. SharePoint policies can apply to entire portals or to any parts thereof, and allow some users to view all webparts on a page while blocking other users from viewing some subset of the webparts on the same page.

### 7.2.5 Required or Recommended Files, Hardware, and Software

<table>
<thead>
<tr>
<th>Component</th>
<th>Required Files</th>
<th>Recommended or Minimum Hardware Requirements</th>
<th>Hardware Used in this Build</th>
<th>Recommended or Minimum Operating System or Other Software</th>
<th>Operating System or Other Software Used in this Build</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Center (CC)</td>
<td>license.dat; ControlCenter-64-7.5.0.0-64-201</td>
<td>1GB RAM; 1GHz CPU; 4GB free disk space</td>
<td>Windows Server 2008, Enterprise Edition, R2, 64-bit, or Windows Server 2012; Java bundled and installed within NextLabs CC; Microsoft SQL Server 2012; Microsoft SQL Server Management Studio</td>
<td>Windows Server 2012; Java bundled and installed within NextLabs software architecture; Microsoft SQL Server 2012; Microsoft SQL Server Management Studio</td>
<td></td>
</tr>
<tr>
<td>External Database</td>
<td>N/A</td>
<td>500 GB for table space</td>
<td>Internal PostgreSQL; External, PostgreSQL, External Oracle, or External MS SQL Server</td>
<td>External MS SQL Server 2012</td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Required Files</td>
<td>Recommended or Minimum Hardware Requirements</td>
<td>Hardware Used in this Build</td>
<td>Operating System or Other Software Used in this Build</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------------</td>
<td>------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Policy Studio</td>
<td>PolicyStudio-7.5.0.0-10-201410291227.zip</td>
<td>i3 or above, 1.5 GHz, dual-core CPU; 2GB; 10 GB free disk space</td>
<td>Windows XP, Service Pack 3, 32-bit, Windows 7, 32-bit and 64-bit, or Windows Server 2008, Enterprise Edition, R2, 64-bit; Microsoft SQL Server 2012; Microsoft SQL Server Management Studio</td>
<td>Windows Server 2012; Microsoft SQL Server 2012; Microsoft SQL Server Management Studio</td>
<td></td>
</tr>
</tbody>
</table>
### Hardware Used in this Build

**Policy Controller**
- 2GB RAM; i3 or above, 1.5 GHz, dual-core CPU; 10 GB free disk space

**Entitlement Manager for SharePoint Server**

### Operating System or Other Software Used in this Build

- Windows Server 2012

### Recommended or Minimum Hardware Requirements

**Policy Controller**
- PolicyController-CE-64-7.0.1.0-1-2-014051916-24.zip

**Entitlement Manager for SharePoint Server**
- SharePointEnforcer-2013-64-7.1.3.0-7-20141-0101427.zip
7.3 Installation and Configuration of NextLabs Control Center (on the SQL Server)

7.3.1 Installation and Configuration

7.3.1.1 Install the Microsoft SQL Server via Microsoft SQLServer 2012

Instructions available at the Microsoft SQLServer site:

Notes

1. Regarding installation of Microsoft SQLServer 2012: if you already completed the
   installation as described in section 4.2.3, this step will already have been completed.

2. Regarding having a database dedicated to NextLabs: NextLabs recommends that for
   anything but a demo or testing environment, you should use a database running on its own
   dedicated server to store all system data, rather than rely on Control Center’s internal
   database. A dedicated database server is strongly recommended because policy
   enforcement data accumulates quickly and can reach a significant volume. The problem is
   not necessarily storage space, but the performance drag on other processes caused by
   database queries of large amounts of data.

7.3.1.2 Create a New Database and Database User for the NextLabs Control Center

Installation and Administration

1. Open Microsoft SQL Server Management Studio and login to Microsoft SQL Server.

2. Right-click on Databases, left-click on New Database.
3. In the New Database window, specify a **Database name** that works for you. The application automatically copies this into the **Logical Names** of the **Database files**. Click **OK**. Example name from this build: **nextlabs**

4. Click on the menu box next to **Security** to begin the process for creating a new login for the new NextLabs database’s administrator.
5. Right-click **Logins**. Left-click **New Login**.

6. Click on **SQL Server authentication**, and enter a new **Login name** and **Password**.

7. Click the menu box next to **Logins**. Right-click on the new user created in the previous step. Click **Properties**.
8. Click on User Mapping, then New Database. Under Database role membership for: [database_name], check the box next to db_owner.
7.3.1.3 Install and Configure the NextLabs Control Center

Complete standard Control Center installation per NextLabs documentation available to customers, using the following steps:

1. Go to your Desktop or other known location where the required NextLabs Control Center installation files are stored. Example:
   \C:\Users\Administrator\Desktop\NextLabs\Platform\7.5.0.0\
   a. Note the location of the required license.dat file which will be needed later; example:
   \C:\Users\Administrator\Desktop\NextLabs\Platform\License\license.dat

2. Right-click on ControlCenter-64-7.5.0.0-64-201410211146.zip and select Extract All from the floating menu. Wait for the files to be extracted.

3. Double-click to open the ControlCenter-64-7.5.0.0-64-201410211146 folder.
4. Right-click on ControlCenterServer-setup.exe, and select Run as administrator.

5. Click Next.
6. Select **I accept the terms in the license agreement**, then click **Next**.

7. Click **Next**.
8. Select the **Complete** setup type. Then, click **Next**.

9. Enter the location of the license file in the **License File Location** field, or click **Change** to navigate to its location in Windows File Explorer. Click **Next**.

Example location: `C:\Users\Administrators\Desktop\Platform\7.5.0.0\ControlCenter-64-7.5.0.0-64-201410211146\license.dat`
10. In the configuration wizard Super User password screen, enter a **Password** for the built-in administrative user for all Control Center Server applications. Click **Next**.

11. At the SSL Certificate Password screen, enter a **Password** to access the SSL certificates for the Control Center Server. Click **Next**.
12. At the Encryption Key Store Password screen, enter a **Password** to access the Encryption Key Store for the Control Center Server. Click **Next**.

13. At the Application User Authentication screen, click **Skip**.
14. At the Control Center Server Database Location screen, select **Store in an external Sql Server database instance**. Click **Next**.

![Control Center Server - Installation](image)

15. At the SQL Server Settings screen, do the following:
   a. Specify the **Connect String**, including the name of the new SQL database created.
      Example: `nextlabs`
   b. Specify **Username** (non-Super User) and **Password**.
   c. Click **Next**. Note: If the error **Connection to the SQL database could not be established properly** appears, it may help to restart the SQL Server.
16. At the Port numbers window, the default port numbers are already entered: Web service port number: 8443, Web application port number: 443. Click **Next**.

17. At the Mail Server Settings screen, click **Skip**.
18. At the Ready to Install the Program screen, click **Install**.

19. At the Installation Wizard Completed screen, click **Finish**.

20. Open an Internet browser and navigate to the following URL: 
   **https://localhost/administrator** to login to the Control Center Administrator web application.
   a. If a security certificate warning comes up, click **Continue to this website**.
   b. Enter the Administrator (Super User) **Username** and **Password**.
21. Once logged into the Control Center Administrator web application in your browser, you can verify that the NextLabs Control Center is installed and configured correctly on the SQL Server, and view the following information:

a. Fully qualified domain name (FQDN) of the server hosting the NextLabs Control Center. Example: SQLServer.ABAC.TEST

b. Services running on the host server, including but not limited to:
   i. Intelligence Server
   ii. Dynamic Access Control
   iii. Key Management Server
   iv. Management Server
   v. Policy Management Server

For more information about these or other services running continuously via NextLabs Control Center on the SQL Server, please refer to NextLabs support documentation.

c. Port via which the above services are running. Example: 8443, default for web services
d. For each of the listed services, the default heartbeat period is 60 minutes, and can be modified via the Administrator (See step 22).

22. Click on the Policy Enforcer Configuration tab. The default Profile to open is the Desktop Enforcer Portal, with the Settings sub-tab defaulted also open. To change the heartbeat frequency for testing or debugging purposes, edit the Heartbeat Frequency field (minimum time is 1 minute). Click Save.
Chapter 7. Setting up NextLabs to Protect SharePoint


7.4.1 Installation

Complete the standard Policy Studio installation per NextLabs documentation available to customers using the following steps:

1. On the SQL Server, go to your Desktop or other known location where the required NextLabs Policy Studio installation files are stored. Example: `C:\Users\Administrator\Desktop\NextLabs\`

2. Right-click on `PolicyStudio-setup64-7.5.0.0-10-201410291227.zip` and select Extract All. Wait for files to be extracted.

3. Double-click to open the `PolicyStudio-setup64-7.5.0.0-10-201410291227` folder.
4. Right-click on PolicyStudio-setup.exe and select Run as Administrator.

5. At the Welcome to the Installation Wizard for Policy Studio screen of the Policy Studio Installation Window, click Next.
6. At the License Agreement screen, select **I accept the terms in the license agreement**, and click **Next**.

7. At the Destination Folder screen, click **Next**.
8. At the Policy Management Server Location screen, enter the default location localhost:8443. Click Next.

9. At the Policy Author Key Store Password screen, enter a Password and click Next.
10. At the Ready to Install the Program screen, click **Install**.

11. At the Installation Wizard Completed screen, click **Finish**.

12. In Windows Explorer, find and open the **policystudio.exe** application file.
   a. Double-click the **C:/ drive**.
   b. Double-click **Program Files**.
   c. Double-click **NextLabs**.
   d. Double-click **Policy Studio**.
e. Double-click `policystudio.exe`.

13. In the Control Center Policy Studio window, enter a **User Name** and **Password** to connect to the Policy Management Server.

14. If the connection is successful, the Control Center Policy Studio - Policy Author window will open.

a. Policies are defined and deployed in this interface, to be covered in chapter 8.
7.5 Installation and Configuration of Policy Controller (PDP)

7.5.1 Installation

To complete standard Policy Controller installation per NextLabs documentation available to customers, use the following steps:

1. On the SharePoint Server, go to your Desktop or other known location where the required NextLabs Policy Controller installation files are stored. Example: C:\Users\Administrator\Desktop\SharePoint\

2. Right-click on PolicyController-CE-64-7.0.1.0-1-201405191624.zip and select Extract All from the floating menu. Wait for files to be extracted.

3. Double-click on PolicyController-CE-64-7.0.1.0-1-201405191624 folder to open it.

4. Double-click CE-PolicyController-setup64.msi to begin installation.

5. At the Welcome to the InstallShield Wizard for NextLabs Policy Controller Installation screen, click Next.
6. At the License Agreement screen, select **I accept the terms in the license agreement** and click **Next**.

7. At the Destination Folder screen, click **Next**.
8. At the ICENet Server Location screen, enter the default ICENet Server Location: \sqlserver:8443. Click Next.

9. At the Ready to Install the Program screen, click Install.
10. At the InstallShield Wizard Completed screen, click **Finish**.

![InstallShield Wizard Completed](image)

11. In the window that immediately opens, click **Yes** to restart the computer, or click **No** to wait and restart after installing the PEP (see section 7.6, Installation and Configuration of NextLabs Entitlement Manager for SharePoint Server).

### 7.6 Installation and Configuration of NextLabs Entitlement Manager for SharePoint Server

#### 7.6.1 Installation and Configuration

Note: Prior to installing the Entitlement Manager for SharePoint Server, it is necessary to install the NextLabs Policy Controller on the SharePoint Server. If you have not already installed the Policy Controller, please refer to section 7.5 before proceeding.

#### 7.6.1.1 Verify that a Web Application Site and Site Collection Already Exist in SharePoint

1. On the SharePoint Server, open an Internet browser and navigate to the following URL: [http://sharepoint:44444/default.aspx](http://sharepoint:44444/default.aspx) to login to the SharePoint Central Administration portal.

2. Enter the **User Name** and **Password** for your SharePoint Central Administration account, and click **OK**.

![Authentication Required](image)
3. At the Central Administration page, click on **Manage web applications** under Application Management.

4. If they do not already exist, create a default **Web Application** site and add it to a basic Site Collection in SharePoint via Central Administration (See Chapter 4).

7.6.1.2 **Install NextLabs Entitlement Manager for SharePoint Server**

Complete the standard Entitlement Manager for SharePoint Server installation per NextLabs documentation available to customers using the following steps:

1. On the SharePoint Server, go to your Desktop or other known location where the required NextLabs Policy Controller installation files are stored. Example: `C:\Users\Administrator\Desktop\SharePoint`.

2. Right-click on `SharePointEnforcer-2013-64-7.1.3.0-7-201410101427.zip` and select **Extract All** from the floating menu. Wait for the files to be extracted.

3. Double-click on the `SharePointEnforcer-2013-64-7.1.3.0-7-201410101427` folder.

4. Double-click on `SharePointEnforcer-2013-64-7.1.3.0-7.msi` to begin the installation.
5. At the Welcome to the InstallShield Wizard for NextLabs Entitlement Manager for Microsoft SharePoint screen, click **Next**.

6. At the License Agreement screen, select **I accept the terms in the license agreement** and click **Next**.
7. At the Ready to Install the Program screen, click **Install**.

8. At the InstallShield Wizard Completed screen, click **Finish**.

9. After installing the IIS server must be reset:
   a. Click on the Windows icon and begin typing the word **PowerShell**
   b. When the Windows PowerShell application icon appears, double-click on the icon to open the Windows PowerShell
   c. From within the Windows PowerShell window, type in this command and press Enter to reset Internet Information Services: **iisreset**

7.6.1.3 **Deploy Entitlement Manager for SharePoint Server to your SharePoint Farm**

On the SharePoint Server, complete standard Entitlement Manager for SharePoint Server deployment per NextLabs documentation available to customers using the following steps:
1. On the SharePoint Server, click the **Start** icon to see the applications pinned to the **Start** menu.

![Start menu](image)

2. Click on the NextLabs Entitlement Manager for SharePoint Server Deployment icon.
   a. This shortcut is automatically pinned during the initial installation. In case the shortcut is not created automatically, the application can be opened from File Explorer at the location: `C:\Program Files\NextLabs\SharePoint Enforcer\bin\NextLabs.Entitlement.Wizard.exe`

3. At the Welcome to NextLabs Entitlement Manager for Microsoft SharePoint Deployment wizard screen, click **Next**.

![Welcome screen](image)
4. At the System Check screen, after the system check is complete, click **Next**.

5. At the Farm Deployment Targets screen, select the applicable web application on which to deploy.
   a. Note: if there is only one entry listed, i.e., `http://sharepoint:44444/Central Administration`, no web applications have been created. In that case, refer back to section 7.6.1.1 or chapter 4.
6. At the Deploying Step 3 of 3 screen, click **Next**.

7. At the Successful Deployment Completed screen, click **Close**.

7.6.1.4 **Enable Policy Enforcement on your Web Application via SharePoint Central Administration**

1. On the SharePoint Server, open an Internet browser and navigate to the following URL: http://sharepoint:44444/default.aspx to login to the SharePoint Central Administration portal.
2. Enter the **User Name** and **Password** for your SharePoint Central Administration account, and click **OK**.

![Authentication Required](image1)

3. Click on the **NextLabs Entitlement Manager** icon.
4. In the page that opens, scroll down to verify that the correct **Web Application** is chosen and the service is **Enabled**.

![Web Application Enabled](image)

### 7.7 Functional Tests

#### 7.7.1 Verify that the NextLabs Webpart for Policy Enforcement has Successfully Been Enabled on the Site Collection in SharePoint

1. Similar to **section 7.6.1.4**, complete the following steps to login to SharePoint Central Administration:
   a. Click on the Start icon.
   b. Click the NextLabs Entitlement Manager for SharePoint icon.
   c. Open SharePoint Central Administration and login as Administrator.
2. Click on **Enable or disable policy enforcement** under the NextLabs Entitlement Manager webpart.

3. Scroll down to the **Web Application** area to verify that the Entitlement Manager is activated for the correct SharePoint web application.
7.7.2 Test to Verify the NextLabs Service is Running

1. Click on the Windows Start icon.
2. Start typing the word Services.
3. Click on the Windows Services icon to open the list of running services.
4. Look for the NextLabs Policy Controller service called Control Center Enforcer Service.
5. Verify that the status is Running.
# Defining Policies and Enforcing Access Decisions with NextLabs

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>Introduction</td>
<td>288</td>
</tr>
<tr>
<td>8.2</td>
<td>Policy Strategy</td>
<td>289</td>
</tr>
<tr>
<td>8.3</td>
<td>Translation of Business Logic into Policy</td>
<td>290</td>
</tr>
<tr>
<td>8.4</td>
<td>Using the NextLabs Policy Studio GUI for Policy Definition</td>
<td>291</td>
</tr>
<tr>
<td>8.5</td>
<td>Configuring Attributes in NextLabs</td>
<td>334</td>
</tr>
<tr>
<td>8.6</td>
<td>Functional Test</td>
<td>339</td>
</tr>
</tbody>
</table>
8.1 Introduction

In previous sections of this How-To Guide, we installed several NextLabs products that can be used to define and deploy Attribute-Based Access Control (ABAC) policies, and enforce decisions regarding user access to Microsoft SharePoint resources based on user, object, and environmental attributes, and the corresponding policies in place. This How-To Guide will illustrate how to use and configure NextLabs Policy Studio, the product responsible for Policy Lifecycle Management, and discuss policy strategy and the translation of business logic into policy.

Within Policy Studio, we will define and deploy policies and policy components. In NextLabs, the word Component is a named definition that represents a category or class of entities, such as users, data resources, or applications; or of actions, such as Open or Copy. Components are similar to using parts of speech to construct policy statements. For example:

Noun: All employees in the human resources department or Any file with an .xls extension
Verb: Copy, Print, or Rename File

Deployment is simply the distribution of new or modified policies and policy components to the appropriate enforcement points on desktop PCs, laptops, and file servers throughout the organization. This means you can create, review and refine policies as long as you like, but they are not enforced until you actually deploy them.

Finally, section 8.6, Functional Test, will illustrate how to ensure that policies are being updated, evaluated, and enforced on Microsoft SharePoint.

8.1.1 Components and Sub-components Used in this How-To Guide

1. NextLabs Policy Studio -provides the Policy Administration Point of the ABAC architecture. This component was installed with the rest of the NextLabs product suite used in this implementation in Chapter 7. Policy Studio provides the graphical user interface for Policy Lifecycle Management (defining, deploying, modifying, and deactivating policies).
   - Located on the SQL Server

2. NextLabs Policy Server SharePoint Enforcer configuration file
   - Automatically exists after NextLabs Control Center installation
   - Located within the NextLabs software architecture on the SQL Server

3. NextLabs AgentLog and bundle.bin files
   - Automatically exist after NextLabs Policy Controller installation
   - Located within the NextLabs software architecture on the SharePoint Server

8.1.2 Pre-requisites to Complete Prior to This How-To Guide

1. If you intend to do a setup without identity federation and federated logins, you must:
   - Install and configure Active Directory (see Chapter 2).
   - Install and configure Microsoft SharePoint (see Chapter 4).
Chapter 8. Defining Policies and Enforcing Access Decisions with NextLabs

8.2 Policy Strategy

8.2.1 Top-level Blacklisting Deny Policy, Whitelisting Allow Sub-policies

In order to demonstrate a policy set with high security and fine-grained control, we employed a general blacklisting, then fine grained whitelisting sub-policy strategy for the policies. We chose this strategy because we considered it a more secure paradigm for securing SharePoint resources. Using this strategy, the access control logic initially applies a general deny all access decision at the top level for a given set of related attributes, then specifies conditions under which access can be allowed in various sub-policies based on sufficient correlating user, resource, and/or environment attributes. For example, later in this guide we will describe a policy set in which we initially deny all users on resources that have a sensitivity level attribute, however there is a sub-policy that specifies that a for resources at sensitivity level 2, allow users with a clearance attribute of Secret during regular business hours. The alternative to this approach would be to apply a general allow all access decision at the top level initially, then specify conditions under which users should be denied access. Because there can be many unforeseen edge cases that may not be anticipated by a business protecting its assets, we consider the general blacklisting, then whitelisting sub-policies approach a more feasibly secure solution. According to our strategy, any time a user, resource, or environment attribute does not comply with a whitelisting sub-policy to allow access, the access decision will default to deny.

8.2.2 Global Policies

In addition to the blacklisting versus “white-listing” approach taken in our policy strategy, we also employed the use of global policies. The term global policy refers to the general applicability of the policy sets to more than one user and more than one resource at a given time. We defined our policies such that they have global effects and do not apply only to very specific use cases by themselves. The collective logic taken from the multiple global policies in place applies to the many kinds of access events that must be controlled according to a business’s complex and distributed business rules, which we describe in section 8.3.
8.3 Translation of Business Logic into Policy

8.3.1 ABAC Build Scenario - Runabout Air Business Rules

In previous sections of our Practice Guide we have constructed an example business scenario where an airline company, Runabout Air, has acquired another airline company, Conway Airlines. In this scenario the two companies have not yet merged their active directory forest and established a trust relationship such that historically Conway Airlines employees will be able to access resources on the Runabout Air SharePoint according to policies that correspond to Runabout Air’s business rules. The business rules we based our policies on are, generally:

1. Some documents are more sensitive than others, and should be marked in SharePoint at different sensitivity levels. These documents should be strictly protected, and access should be restricted to Runabout Air’s normal business hours. Also, users should only be granted access to sensitive documents if they have sufficient clearance.

2. Users should only be able to access documents that belong to their department, or to the departments relevant to them in the case of some instances of a need for cross-department access, i.e., business intelligence employees should have access to both sales and marketing department documents.

3. Some documents are time-sensitive and pertain to system or other business maintenance, and should be marked in SharePoint as maintenance documents. These documents should only be accessed outside of Runabout Air’s normal business hours, so as to reduce the likelihood of disruption of normal business operation.

4. There are times when a suspicious IP address or range of addresses should be blocked from accessing any SharePoint resources, or when a user from a particular IP address or range of IP addresses should only have access to low-sensitivity documents. There must be a mechanism in place to ensure access is denied for users attempting to access any high-sensitivity documents from an environment with that IP address or within a given IP address range.

8.3.2 Translation of Runabout Air Business Rules into ABAC Policies

ABAC Policies created from the above business rules might look like this:

1. Top-level sensitivity policy: default to deny access to all users attempting to access resources that have a sensitivity level attribute defined in SharePoint as greater than 0, unless explicitly allowed access by a sub-policy.
   a. For documents whose sensitivity attribute is defined as 1, allow access any time of day, any day of the week, to users with a clearance attribute of None, Secret, or Top Secret.
   b. For documents whose sensitivity attribute is defined as 2, allow access between the hours of 6am and 6pm for users with a clearance attribute of Secret or Top Secret.
   c. For documents whose sensitivity attribute is defined as 3, allow access between the hours of 6am and 6pm for users with a clearance attribute of Top Secret.

2. Top-level department policy: default to deny access to all users attempting to access resources that have a department attribute and project status defined in SharePoint.
a. For users whose department attribute is defined as a value equal to the document's department attribute value, allow access for documents with a project status of any value.

b. For users whose department attribute is Business Intelligence, allow access for documents with a department attribute of Sales or Marketing and with a Project status of any value.

c. Note: The Project status metric is necessary because the department attribute is defined at the site level within SharePoint. Restricting users based only on the resource's department attribute in this policy set results in the user being stuck in a deny access loop, no longer being able to access the Runabout Air root site and navigate to their correct department's documents. Because each document has a project status attribute defined in addition to the department attribute, the policies can specify the targets of this policy as having both project status and department attributes defined, even though the department attribute is the most pertinent attribute for enforcing the access control relating to department access rules.

3. Top-level maintenance policy: default to deny access to all users attempting to access resources that have a maintenance attribute defined in SharePoint.
   a. For documents whose maintenance attribute is defined as no, allow access to users, any time of day, any day of the week.
   b. For documents whose maintenance attribute is defined as yes, allow access to users between 6pm and 6am, any day of the week.

4. Top-level IP Address policy: default to deny access to all users attempting to access resources that have a sensitivity attribute defined in SharePoint.
   a. For documents whose sensitivity attribute is defined as 1, allow access to any user from an environment with any IP address defined.
   b. For documents whose sensitivity attribute is defined as 2 or 3, allow access to users coming from an environment with an IP address other than a restricted IP or one within a restricted IP range.

8.4 Using the NextLabs Policy Studio GUI for Policy Definition and Deployment

In this section we will provide step-by-step instructions for how to define, deploy, modify and re-deploy, and deactivate necessary policy components and policies within Policy Studio. The examples we will use correspond to the Runabout Air business rules and ABAC policies described in section 8.3.1 and section 8.3.2. Note that Policy Studio was installed on the SQL Server, which is where all of the activity in section 8.4 occurs.
8.4.1 Login and Initial Screen in Policy Studio

Given you have followed the instructions found in chapter 7, follow these instructions to login to the NextLabs Policy Studio:

1. In Windows Explorer, find and open the `policystudio.exe` application file:
   a. Double-click the `C:`/ drive.
   b. Double-click `Program Files`.
   c. Double-click `NextLabs`.
   d. Double-click `Policy Studio`.
   e. Double-click `policystudio.exe`.

2. In the Control Center Policy Studio window, enter **User Name** and **Password**, then click **Login** to connect to the Policy Management Server.

3. If login was successful, you will see the Policy Studio's graphical user interface, specifically the main screen where new policies and new components are defined, deployed, modified, and deactivated. Note the **Policies** panel in the top-left, the **Components** panel in the bottom-left, and an open space to the right where editing panels emerge for editing the policies and components.
4. After following the instructions in this section to define and deploy several user and resource components, as well as four policy sets, the Policy Studio interface will show the new components and policies populated in the left-side panel.
8.4.2 Policy Studio Menu Commands

Below are some of the Policy Studio menu commands used in this How-To Guide, along with explanations for what action they perform.

Extracted from the NextLabs Policy Studio User guide available to customers:

<table>
<thead>
<tr>
<th>Menu</th>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Exit</td>
<td>Closes Policy Studio.</td>
</tr>
<tr>
<td>Edit</td>
<td>Delete</td>
<td>Deletes the currently selected item or items.</td>
</tr>
<tr>
<td></td>
<td>Duplicate</td>
<td>Creates a clone of the selected component.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Menu</th>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions</td>
<td>Modify</td>
<td>Changes the status of the currently displayed component or policy to Draft. You must do this whenever you want to make any changes to a component or policy that has been submitted. Function is the same as the Modify button at the bottom of the Editing pane.</td>
</tr>
<tr>
<td></td>
<td>Submit</td>
<td>Submits the currently selected components or policies for changing from one status to another—for example, from Draft status to Submitted for Deployment. Function is the same as the Submit button at the bottom of the Editing pane. Disabled if no object is selected, or if any of the selected objects is not currently in Modify state.</td>
</tr>
<tr>
<td></td>
<td>Deploy</td>
<td>Deploys the currently displayed component or policy. Function is the same as the Deploy button at the bottom of the Editing pane. As with individually deployed objects, you can specify a scheduled deployment, or choose how. Disabled if no object is selected, or if the selected object has not been submitted for deployment.</td>
</tr>
<tr>
<td></td>
<td>Deploy All</td>
<td>Deploys all currently submitted components or policies. Function is the same as the Deploy button at the bottom of the Editing pane.</td>
</tr>
<tr>
<td></td>
<td>Deactivate</td>
<td>Changes the status of the currently selected policies or components from Active to Deactivated. Disabled if no object is selected, or if any of the selected objects is not currently in Active state.</td>
</tr>
<tr>
<td>Window</td>
<td>Preview</td>
<td>Opens the Preview pane, at the right side of the Editor pane. The Preview pane allows you to test the actual content that would result from the current definition of a component.</td>
</tr>
<tr>
<td></td>
<td>Policy Manager</td>
<td>Toggles to the Policy Manager interface. You can also type Ctrl + Tab.</td>
</tr>
<tr>
<td></td>
<td>Policy Author</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

8.4.3 Defining and Deploying Components

8.4.3.1 Explanation of Components in NextLabs

According to the NextLabs Policy Studio User Guide available to customers, it is necessary to define components to represent various kinds of entities in your information environment. There are several times when you might want to define a new component:

1. After setting up your Control Center system, before constructing policies for the first time (which is the reason here at this point in our How-To literature)
2. When new classes of information or users come under the control of information policy
3. When a new policy requires a policy component that has not yet been created
4. When conditions at the organization change in any way that adds new items to be covered by information control policies. For example, if the company reorganizes and adds a new division, you might need a new policy component to represent the employees in that division.

Furthermore, when you are constructing a component, you do not need to save your work explicitly. Work is automatically saved as you go. If you are interrupted while working on a policy component, or want to work on another task and return to constructing the policy component later, you can stop and continue the constructing process as desired. Your work will
be saved in draft status. You can find the policy component later in the appropriate component panel.

8.4.3.2 Defining and Deploying User Components

According to the Runabout Air business rules in section 8.3.2 and ABAC policies in section 8.3.2, it is possible that you may need to create a User Component to match the following conditions: user clearance attribute, user department attribute, and user IP address. This is correct except for the user department attribute. Because of the cross-departmental access of Runabout Air’s Business Intelligence employees, we use logical syntax instead of graphical components while defining that policy. Also, a note regarding the user IP address component: even though IP address is an environmental attribute, it can be configured in NextLabs as a user attribute coming from SharePoint Claims, or as a resource attribute, which requires different configuration in NextLabs. For our example we use the IP Address from SharePoint Claims, which is handled as a user attribute.

8.4.3.2.1 Clearance Components

**Clearance = None**

1. In the Components panel in the bottom-left of the Policy Studio window, click on the **Subjects** heading, and then click on the **Users** tab. Then click **New** to create a new component.

2. In the Create New User Component window, enter a descriptive component name, such as **clearance = None**. Click OK.
3. In the component editing panel you will see the following:

![Component Editing Panel]

4. In the editing panel, click on the plus sign box under Property Name and enter clearance in the property name text box, keep the default is as the action, then enter None into the value text box. Click Submit.
5. In the Submit window, click **Submit**.

6. From the component editing panel, note the differences. The new status reads **Submitted for Deployment**. Click **Deploy**.
7. In the Deploy window, click **OK**. Note: You may deploy immediately, which we choose in our example. You could also deploy the following day at midnight, or at a different specific date and time.

8. Verify at the bottom of the component editing panel that the Status now reads **Pending Deployment**. This will remain for the duration of the heartbeat (described in chapter 7).

9. After the duration of the heartbeat has passed, Status will then read as **Deployed**. This indicates that the component is actively deployed in your ABAC system.

**Clearance = Secret**

The easiest way to create additional attribute components is to duplicate existing ones. To duplicate the existing user attribute component:

1. From the Component panel, highlight the name of the existing component, i.e., **clearance = None**
2. Click on **Edit** from the menu toolbar at the top of the window and select **Duplicate** from the drop-down menu, or right-click on the component and select **Duplicate** from the floating menu:

3. In the Duplicate window, edit the name of the new component, i.e., **clearance = Secret**. Click **Save**.
4. Edit the property value to match the component's purpose, i.e., **Secret**. Click **Submit**.

5. Repeat steps 5-9 from Clearance = None to Submit and Deploy this component.

**Clearance = Top Secret**
1. Repeat steps 1-5 in Clearance = Secret for duplicating a new user attribute component. The new component should be named clearance = **Top Secret**, and the property value should equal **Top Secret**.

8.4.3.2.2 IP Address Component
1. Repeat steps 1-3 in Clearance = Secret for duplicating a new user attribute component. The new component should be named **ip_address = 10.33.7.211**.
2. From the component editing panel, edit the Property Name to **ip_address** and the value to **10.33.7.211**, leaving the default action is. Then click **Submit**.

3. Repeat steps 5-9 from the **Clearance = None** to Submit and Deploy this component.

### 8.4.3.3 Defining and Deploying Resource Components

#### 8.4.3.3.1 Maintenance Components

**Maintenance = yes**

1. In the Components panel in the bottom-left of the Policy Studio window, click on the **Resources** heading, and then click on the **Portals** tab. Then, click **New** to create a new component.
2. Enter a descriptive component name, such as `maintenance = yes`, then click **OK**.

3. In the editing panel, click on the **plus sign** box under Property Name and enter `maintenance` in the **Property Name** text box, keep the default is as the action, and enter `yes` into the value text box. Then click **Submit**.

4. Repeat steps 5-9 from **Clearance = None** to **Submit and Deploy this component**.
**Maintenance = no**

Similar to the steps taken for duplicating user components, do the following to duplicate the existing resource maintenance component to create the other resource components.

1. In the Component panel in the bottom-left corner of the Policy Studio interface, right-click on the `maintenance = yes` component. In the floating menu, select **Duplicate**.

2. In the Duplicate window, edit the name of the new component. Example: `maintenance = no`. 
3. In the component editing panel, change the property value to no and click Submit.

4. Repeat steps 5-9 from Clearance = None to Submit and Deploy this component.

8.4.3.3.2 Sensitivity components

**Sensitivity = 1**

1. Repeat steps 1-4 from Maintenance = no to duplicate an existing resource component to create the Sensitivity = 1 component.

**Sensitivity = 2**

1. Repeat steps 1-4 from Maintenance = no to duplicate an existing resource component to create the Sensitivity = 2 component.

**Sensitivity = 3**

1. Repeat steps 1-4 from Maintenance = no to duplicate an existing resource component to create the Sensitivity = 3 component.
8.4.3.3 Project status component

**Project status = any**

1. Repeat steps 1-4 from Maintenance = no to duplicate an existing resource component to create the Project status = any component.

2. **Note:** Before the Submit step, in the component editing panel, enter the property value as 

![Portal Component](image)

8.4.4 Defining Policy

After following the steps to define and deploy components in section 8.4.3, you can continue on to define policies that relate to the Runabout Air scenario business rules discussed in section 8.3. In order to define policies in Policy Studio, login as described in section 8.4.1.

8.4.4.1 Creating a Policy Set Folder

Before being able to create any policies in Policy Studio, first you must create a folder, or choose an existing one.
1. From the main Policy Studio window, click **New Folder**.

2. Enter the **name** of your folder and click **OK**.
8.4.4.2 Defining Department-based Policy Set

8.4.4.2.1 Defining the Top-level Department Policy that Enforces a General Deny Decision

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new folder to highlight it. Then click New Policy.

2. In the Create New Policy window, enter a name for the new policy. From the Policy Type drop-down menu, select Document Policy (which applies to all SharePoint policies). Click OK.

3. The new policy opens automatically in an editing panel. For this policy, keep the default Deny enforcement. Make these edits:

   a. In the On Resources area, click on the plus sign box next to Target. This automatically populates in and Resource Component.

   b. In the Condition Expression enter the ACPL: (resource.portal.department = "*" AND resource.portal.project status = "*")
c. In the Obligations area, check the **Display User Alert** box in order to customize the deny message displayed to the user when access is denied.

4. In the policy editing panel, your policy should look like this:

![Policy Editing Panel Image]

5. To deploy this policy, follow the steps in section 8.4.5.

### 8.4.4.2.2 Defining a Department-based Sub-policy that Enforces an Allow Decision when Certain Conditions are met

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new policy to highlight it. Then click on **New Policy** to create a sub-policy.

2. Select a **name** for the new sub-policy then click **OK**.
3. In the policy editing panel, make the following edits:
   a. From the Enforcement drop-down menu, select *Allow*.

   ![Document Policy](image)

   b. In the On Resources area, click on the plus sign box next to *Target*.
      i. In the Components panel, click on *Resources*, then the *Portals* tab to see the components you created earlier.

   ![Components Panel](image)

      ii. From the Portals tab, left-click and hold the *Project status = any* component and drag it onto the *Target* field.

   ![Target Field](image)

   c. In the Conditions area, in the *Condition Expression* text box, enter the ACPL:
      (user.department = resource.portal.department OR (user.department = "Business")
4. In the Policy Editing panel, your policy should look like this:

5. To deploy this policy, follow the steps in section 8.4.5.
8.4.4.3 Defining a Sensitivity-based Policy Set

In order to define a sensitivity-based policy set, follow instructions similar to defining the department-based policy set in section 8.4.4.2:

8.4.4.3.1 Defining the Top-level Sensitivity Policy that Enforces a General Deny Decision

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your folder to highlight it. Then click on New Policy.

2. In the Create New Policy window, enter a name for the new policy. From the Policy Type drop-down menu, select Document Policy (which applies to all SharePoint policies). Click OK.

3. The new policy opens automatically in an editing panel. For this policy, keep the default Deny enforcement. Make these edits:
   a. In the On Resources area, click on the plus sign box next to Target. This automatically populates in and Resource Component.
   b. In Condition Expression enter the ACPL: resource.portal.sensitivity > "0"
4. In the Obligations area, check the **Display User Alert** box in order to customize the deny message displayed to the user when access is denied.

![Obligations panel](image)

5. In the policy editing panel, your policy should look like this:

![Document Policy panel](image)

6. To deploy this policy, follow the steps in section 8.4.5.
8.4.3.2 Defining a Sensitivity-based Sub-policy that Enforces an Allow Decision when Certain Conditions are Met for Access to Sensitivity Level 1 Documents

Similar to the steps in section 8.4.4.2.2 for creating the Department-based sub-policy, do the following:

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new policy to highlight it. Then click New Policy to create a sub-policy.

2. Select a name for the new sub-policy then click OK.

3. In the policy editing panel, make the following edits:
   a. From the Enforcement drop-down menu, select Allow.
   b. In the Subject area, click on the plus sign next to User.
      i. In the Components panel in the bottom-left corner of the Policy Studio window, click on Subjects, then the Users tab to see the components you created earlier.
      ii. Left-click and hold the clearance = None component to drag it onto the User field.
      iii. Left-click and hold the clearance = Secret component to drag it onto the User field.
      iv. Left-click and hold the clearance = Top Secret component to drag it onto the User field.
   c. In the On Resources area, click on the plus sign box next to Target.
      i. In the Components panel in the bottom-left corner of the Policy Studio window, click on Resources, then the Portals tab to see the components you created earlier.
      ii. Left-click and hold the sensitivity = 1 component to drag it onto the Target field.
   d. In the policy editing panel, your policy should look like this:
e. To deploy this policy, follow the steps in section 8.4.5.

8.4.4.3.3 Defining a Sensitivity-based Sub-policy that Enforces an Allow Decision when Certain Conditions are Met for Access to Sensitivity Level 2 Documents

Similar to the steps in section 8.4.4.3.2 for creating the sensitivity-based sub-policy for sensitivity level 1 documents, do the following:

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new policy to highlight it. Then click **New Policy** to create a sub-policy.

2. Select a name for the new sub-policy then click **OK**.

3. In the policy editing panel, make the following edits:
Chapter 8. Defining Policies and Enforcing Access Decisions with NextLabs

a. From the **Enforcement** drop-down menu, select **Allow**.

b. In the **Subject** area, click on the **plus sign** next to User.
   
i. In the Components panel in the bottom-left corner of the Policy Studio window, click on **Subjects**, then the **Users** tab to see the components you created earlier.

   ![Subjects Panel](image.png)

   ii. Left-click and hold the **clearance = Secret** component to drag it onto the **User** field.

   iii. Left-click and hold the **clearance = Top Secret** component to drag it onto the **User** field.

c. In the **On Resources** area, click on the **plus sign** box next to **Target**.

   i. In the Components panel in the bottom-left corner of the Policy Studio window, click on **Resources**, then the **Portals** tab to see the components you created earlier.

   ii. Left-click and hold the **sensitivity = 2** component to drag it onto the **Target** field.
d. In the Conditions area, click on the plus sign boxes next to Time and Day. Edit those fields to match below:

![Conditions diagram]

4. In the policy editing panel, your policy should look like this:
5. To deploy this policy, follow the steps in section 8.4.5.

### 8.4.4.3.4 Defining a Sensitivity-based Sub-policy that Enforces an Allow Decision when Certain Conditions are Met for Access to Sensitivity Level 3 Documents

Similar to the steps in section 8.4.4.3.2 for creating the sensitivity-based sub-policy for sensitivity level 1 documents, do the following:

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new policy to highlight it. Then click **New Policy** to create a sub-policy.
2. Select a name for the new sub-policy then click **OK**.
3. In the policy editing panel, make the following edits:
   a. From the **Enforcement** drop-down menu, select **Allow**.
   b. In the Subject area, click on the **plus sign** next to User.
      i. In the Components panel in the bottom-left corner of the Policy Studio window, click on **Subjects**, then the **Users** tab to see the components you created earlier.
      ii. Left-click and hold the **clearance = Top Secret** component to drag it onto the User field.
   c. In the On Resources area, click on the **plus sign** box next to Target.
      i. In the Components panel in the bottom-left corner of the Policy Studio window, click on **Resources**, then the **Portals** tab to see the components you created earlier.
      ii. Left-click and hold the **sensitivity = 3** component to drag it onto the Target field.
d. In the Conditions area, click on the **plus sign** boxes next to **Time** and **Day**. Edit those fields to match below:

![Diagram showing Conditions section with Time and Day settings]

4. In the policy editing panel, your policy should look like this:

![Policy editing panel with various fields and settings]
5. To deploy this policy, follow the steps in section 8.4.5.

8.4.4.4 Defining a Maintenance-based Policy Set

In order to define a maintenance-based policy set, follow instructions similar to defining the department-based policy set in section 8.4.4.2:

8.4.4.4.1 Defining the Top-level Maintenance Policy that Enforces a General Deny Decision

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new folder to highlight it. Then click New Policy.

2. In the Create New Policy window, enter a name for the new policy. From the Policy Type drop-down menu, select Document Policy (which applies to all SharePoint policies). Click OK.

3. The new policy opens automatically in an editing panel. For this policy, keep the default Deny enforcement. Make these edits:

   a. In the On Resources area, click on the plus sign box next to Target. This automatically populates in and Resource Component.

   b. In Condition Expression, enter the ACPL: resource.portal.maintenance = "*"

   c. In the Obligations area, check the Display User Alert box in order to customize the deny message displayed to the user when access is denied.
4. In the policy editing panel, your policy should look like this:

![Policy Editing Panel](image)

5. To deploy this policy, follow the steps in section 8.4.5.

8.4.4.2 Defining a Maintenance-based Sub-policy that Enforces an Allow Decision when Certain Conditions are Met for Access to Documents whose Maintenance Attribute is defined as Yes

Similar to the instructions in section 8.4.4.2 for defining a Department-based sub-policy, do the following:

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new policy to highlight it. Click **New Policy** to create a sub-policy under this main policy.

2. **Select a name** for the new sub-policy, then click **OK**.

3. In the policy editing panel, make the following edits:
   a. From the **Enforcement** drop-down menu, select **Allow**.
   b. In the On Resources area, click on the **plus sign** box next to **Target**.
i. In the Components panel in the bottom-left corner of the Policy Studio window, click on Resources, then the Portals tab to see the components you created earlier.

ii. Left-click and hold the maintenance = yes component to drag it onto the Target field.

c. In the Conditions area, click on the plus sign boxes next to Time and Day. Edit those fields to match below:

![Conditions Diagram]

4. In the policy editing panel, your policy should look like this:
5. To deploy this policy, follow the steps in section 8.4.5.

8.4.4.3 Defining a Maintenance-based Sub-policy that Enforces an Allow Decision when Certain Conditions are Met for Access to Documents whose Maintenance Attribute is defined as No

Similar to the instructions in section 8.4.4.2.2 for defining a Department-based sub-policy, do the following:

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new policy to highlight it. Click **New Policy** to create a sub-policy.

2. Select a name for the new sub-policy, then click **OK**.

3. In the policy editing panel, make the following edits:
   a. From the **Enforcement** drop-down menu, select **Allow**.
   b. In the On Resources area, click on the **plus sign** box next to **Target**.
      i. In the Components panel in the bottom-left corner of the Policy Studio window, click on **Resources**, then the **Portals** tab to see the components you created earlier.
ii. Left-click and hold the **maintenance = no** component to drag it onto the **Target** field.

4. In the policy editing panel, your policy should look like this:

![Document Policy]

5. To deploy this policy, follow the steps in section 8.4.5.

8.4.4.5 Defining an IP Address-based Policy Set

In order to define an IP address-based policy set, follow instructions similar to defining the department-based policy set in section 8.4.4.2:

8.4.4.5.1 Defining the top-level IP Address Policy that Enforces a General Deny Decision

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new folder to highlight it. Then click **New Policy**.

2. In the Create New Policy window, enter a **name** for the new policy. From the **Policy Type** drop-down menu, select **Document Policy** (which applies to all SharePoint policies). Click **OK**.
3. The new policy opens automatically in an editing panel. For this policy, keep the default \textit{Deny} enforcement. Make these edits:
   a. In the \textbf{Condition Expression}, enter the ACPL: \texttt{resource.portal.sensitivity = "*"}
   b. In the Obligations area, check the \textbf{Display User Alert} box in order to customize the deny message displayed to the user when access is denied.
4. In the policy editing panel, your policy should look like this:

![Policy Editing Panel](image)

5. To deploy this policy, follow the steps in \textit{section 8.4.5}.

\textbf{8.4.4.5.2} Defining an IP Address-based Sub-policy that Enforces an Allow Decision for Access to Resources at any Sensitivity Level when a User Does not Come from an Environment with a Restricted IP Address (ex: 10.33.7.211)

Similar to the instructions in \textit{section 8.4.4.2.2} for defining a Department-based sub-policy, do the following:

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new policy to highlight it. Click \textbf{New Policy} to create a sub-policy.
2. Select a name for the new sub-policy, then click OK.

3. In the policy editing panel, make the following edits:
   a. From the Enforcement drop-down menu, select Allow.
   b. In the On Resources area, click on the plus sign box next to Target.
      i. In the Components panel in the bottom-left corner of the Policy Studio window, click on Resources, then the Portals tab to see the components you created earlier.
      ii. Left-click and hold the sensitivity = 1 component to drag it onto the Target field.

4. In the policy editing panel, your policy should look like this:

5. To deploy this policy, follow the steps in section 8.4.5.
8.4.4.5.3 Defining an IP Address-based Sub-policy that Enforces an Allow Decision for Access to Resources at Only Sensitivity Level 1 when a User Comes from an Environment with a Restricted IP Address (ex: 10.33.7.211)

Similar to the instructions in section 8.4.4.2.2 for defining a Department-based sub-policy, do the following:

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on your new policy to highlight it. Then click New Policy to create a sub-policy.

2. Select a name for the new sub-policy, then click OK.

3. In the policy editing panel, make the following edits:
   a. From the Enforcement drop-down menu, select Allow.
   b. In the Subject area, click on the plus sign box next to User.
      i. From the drop-down menu, select not in.
      ii. In the Components panel in the bottom-left corner of the Policy Studio window, click on Subjects, then the Users tab to see the components you created earlier.
      iii. Left-click and hold the ip_address=10.33.7.211 component to drag it onto the User field.

   c. In the On Resources area, click on the plus sign box next to Target.
      i. In the Components panel in the bottom-left corner of the Policy Studio window, click on Resources, then the Portals tab to see the components you created earlier.
      ii. Left-click and hold the sensitivity = 1 component to drag it onto the Target field.
      iii. Left-click and hold the sensitivity = 2 component to drag it onto the Target field.
      iv. Left-click and hold the sensitivity = 3 component to drag it onto the Target field.
4. In the policy editing panel, your policy should look like this:

![Policy Example]

5. To deploy this policy, follow the steps in section 8.4.5.
8.4.5 Deploying Policy

In order to deploy policies, follow steps similar to those for deploying a component (see the section Clearance = None):

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on the policy you want to deploy. In the policy editing panel, click Submit.

   ![Policy Editing Panel]

   a. Or, in the Policies panel in the top-left corner of the main Policy Studio window, right-click the policy you want to deploy. Select Submit from the floating menu.

   ![Policy List Panel]
2. In the Submit window, click **Submit**.

![Submit window](image)

3. From the component editing panel, note the differences. The new status reads **Submitted for Deployment**. Click **Deploy**.
   
a. Or, in the Policies panel in the top-left corner of the main Policy Studio window, right-click the policy you want to deploy. Select **Deploy** from the floating menu.
4. In the Deploy window, click **OK**. Note: You may specify to deploy immediately, which we choose in our example. You may also deploy at the following day at midnight, or at a different specific date and time.

5. At the bottom of the policy editing panel, verify that the **Status** is now **Pending Deployment**. This will remain for the duration of the heartbeat (described in chapter 7).

6. After the duration of the heartbeat has passed, **Status** should read as **Deployed**. This indicates that the component is actively deployed in your ABAC system.

8.4.6 Modifying and Re-Deploying Policies and Components

In order to modify existing policies and re-deploy them, do the following:

8.4.6.1 Modifying and Deploying Existing Policies

1. In the Policies panel in the top-left corner of the main Policy Studio window, click on the policy you want to modify. In the policy editing panel, click **Modify**.
   a. Or, right-click the policy you want to modify and select **Modify** from the floating menu.

2. In the policy editing panel, make the desired changes and click **Submit**.

3. Follow the deploy instructions from section 8.4.5 to deploy the modified policy.

8.4.6.2 Modifying and Deploying Existing Components

1. In the Components panel in the bottom-left corner of the main Policy Studio window, click on the component you want to modify. In the policy editing panel, click **Modify**.
   a. Or, right-click the component you want to modify and select **Modify** from the floating menu.
2. In the component editing panel, make the desired changes and click **Submit**.

3. Follow the deploy instructions from section 8.4.5 to deploy the modified component.

### 8.4.7 Deactivating Policies and Components

#### 8.4.7.1 Deactivating Policies

1. In the Policies panel in the top-left corner of the main Policy Studio window, right-click the policy you want to deactivate. Select **Deactivate** from the floating menu.

2. At the bottom of the policy editing panel, note the change in **Status** to **Pending Deactivation**. Click **Deploy**.
3. In the Deploy window, click **OK**. Note: You may specify to deploy immediately, which we choose in our example. You may also deploy the following day at midnight, or at a different specific date and time.

4. Verify at the bottom of the policy editing panel that the **Status** is now **Pending Deactivation**. This will remain for the duration of the heartbeat (described in chapter 7).

5. After the duration of the heartbeat has passed, **Status** should read as **Inactive**. This indicates that the component is currently inactive in your ABAC system.

### 8.4.7.2 Deactivating Components

1. In the Components panel in the bottom-left corner of the main Policy Studio window, right-click on the component you want to deactivate. Select **Deactivate** from the floating menu.

2. Follow steps 2-5 in **section 8.4.7.1** for deactivating policies.

### 8.4.8 Deleting Policies and Components

**Note:** To delete a policy or component, you must first deactivate the item and any related sub-items.
8.4.8.1 Deleting Policies

1. In the Policies panel in the top-left corner of the main Policy Studio window, right-click on the policy you want to delete. Select **Delete** from the floating menu.

2. In the Delete window, click **Yes**.

8.4.8.2 Deleting Components

1. In the Components panel in the bottom-left corner of the main Policy Studio window, right-click on the policy you want to delete. Select **Delete** from the floating menu.

8.5 Configuring Attributes in NextLabs

Chapter 6 illustrates how to configure the attribute flow between several of the servers and components in the ABAC architecture. Note that the NextLabs Entitlement Manager was installed on the SharePoint Server, which is where all of the activity in section 8.5 occurs.

In order to configure NextLabs to enforce policy on all of the attributes coming from the front-channel as SharePoint Claims, you must first stop the NextLabs Policy Controller service, edit the configuration.xml file in the SharePoint Enforcer software architecture, restart Internet Information Services (IIS), then restart the NextLabs Policy Controller service using the following instructions.

8.5.1 Stopping the NextLabs Policy Controller Service

1. On the SharePoint Server, click the Windows icon and begin typing the word **Services**.
2. Double-click on the icon to open the Services application.
3. Within the Services application window, in the list of services, click on the **Name** column to sort by alphabetical order, and look for **Control Center Enforcer Service**.
4. If the status of the Control Center Enforcer Service is **Running**, stop it.
   a. Click the Windows icon.
b. Double-click the **Stop Policy Controller** shortcut icon.

c. Enter your NextLabs Administrator credentials. Then click **Stop**.

d. In the Stop Enforcer Service success window, click **OK**.

### 8.5.2 Editing the Configuration File

#### 8.5.2.1 Locating and Opening the SharePoint Enforcer configuration.xml File

1. In Windows Explorer, find and open the SharePoint Enforcer configuration.xml file.
   a. Double-click the **C:/** drive.
   b. Double-click **Program Files**.
   c. Double-click **NextLabs**.
   d. Double-click **SharePoint Enforcer**.
   e. Double-click **config**.
   f. Right-click **Configuration.xml** to edit the file in a text editor.
8.5.2.2 Configuring Resource Attributes from SharePoint Metadata

1. Within the configuration.xml file, look for the <SPEConfiguration> tag.

2. Under that tag, but above a <User Attribute> tag, insert tags for each site-level or sub-site level resource attribute of interest.
   a. For example, in our build we created policies based on the department resource attribute, so in our configuration.xml file we included the following:

   <PropertyBag disabled="false" level="SiteCollection">
   <Property disabled="false" name="department" attributename="department" />
   </PropertyBag>

   <PropertyBag disabled="false" level="SubSite">
   <Property disabled="false" name="department" attributename="department" />
   </PropertyBag>

   b. From the example above, the top of the configuration.xml file looks like this:

```
<PropertyBag disabled="false" level="SiteCollection">
  <Property disabled="false" name="department" attributename="department" />
</PropertyBag>

<PropertyBag disabled="false" level="SubSite">
  <Property disabled="false" name="department" attributename="department" />
</PropertyBag>
```

8.5.2.3 Configuring User Attributes from SharePoint Claims

1. Within the configuration.xml file directly under any <PropertyBag> closing tags, find the <User Attribute> </User Attribute> portion of the document. Initially, its default contents
in that area may look like this, containing some default user attributes such as "emailAddress" or "adfsGroup":

2. In the User Attribute area, add more claims here to include all the attributes you will be expecting to evaluate in NextLabs policies for access control decisions.
   a. For example, in our build we created policies based on users' "clearance", "department", and "ip_address", so in our configuration.xml file we included the following, among others:

   ```xml
   <Claim name="department" attributename="department" claimtype="http://schemas.xmlsoap.org/ws/2005/05/identity/claims /department" disabled="false" />
   <Claim name="ip_address" attributename = "ip_address" claimtype="http://schemas.xmlsoap.org/ws/2005/05/identity/claims /ip_address" disabled="false" />
   <Claim name="clearance" attributename = "clearance" claimtype="http://schemas.xmlsoap.org/ws/2005/05/identity/claims /clearance" disabled="false" />
   
   From the example above, the rest of our configuration.xml file looks like this:
8.5.2.4 Saving Changes to the Configuration File

1. From the File menu, click Save, or Ctrl+S on your keyboard.

1. Click the Windows icon.
2. In the Search text box, begin typing PowerShell.
3. Click on Windows PowerShell.
4. In the PowerShell window, type the command: `iisreset`. Press Enter.
5. In the PowerShell window, verify that services stopped and restarted successfully.

8.5.4 Restarting the NextLabs Policy Controller Service

1. Click on the Windows icon and begin typing the word Services.
2. Double-click the Services icon to open the application.
3. Within the Services application window in the list of services, click on the Name column to sort by alphabetical order and look for Control Center Enforcer Service.
4. Right-click Control Center Enforcer Service and click Start.
   a. It may be necessary to click the Refresh icon in order to see the Control Center Enforcer Service status change to Running.

8.6 Functional Test

8.6.1 Updated bin file after Policy Creation/modification

After a policy or component is deployed for the first time, or modified and re-deployed within Policy Studio on the SQL Server, an encrypted bundle.bin file on the SharePoint Server will be updated after one heartbeat. As explained in chapter 7, on the SharePoint Server it is the responsibility of the Controller Manager component of the NextLabs Policy Controller (PDP) to encrypt the bundle.bin file on the local file system for use during policy evaluation by the PDP.

To ensure the policy logic is being correctly sent from the NextLabs Policy Studio (PAP) on the SQL Server to the bundle.bin file on the SharePoint Server for use by the NextLabs Policy Controller (PDP), you can find the bundle.bin file and decrypt its contents to see your policy logic decrypted there.

8.6.1.1 On the SharePoint Server note timestamp of the bundle.bin file and decrypt its contents

1. Double-click the C:/ drive.
2. Double-click Program Files.
3. Double-click NextLabs.
4. Double-click Policy Controller.
5. Scroll down to find `bundle.bin` and note the timestamp in the **Date Modified** column. This would be the last time policies or components were deployed.

6. Scroll back up and double-click on the `bin` folder.
7. Scroll down to find **Decrypt.exe**.

![ Decrypt.exe window ](image1)

a. In the Decrypt window, enter the administrator’s **Password** and press **Enter**.

![ Decrypt window ](image2)

b. After the Decrypt window disappears, click on **Policy Controller** to return to that folder. Scroll down and double-click the **bundle.out** file.

![ Policy Controller window ](image3)
c. In the text editor window, scroll down to find policies that you have created previously. Example: RunaboutAirPolicySets/SharePoint Protection - Department top-level policy

8.6.2 Reviewing NextLabs AgentLog to Illustrate History of Access Control Evaluations During SharePoint Access

1. Double-click the C:/ drive.
2. Double-click Program Files.
3. Double-click NextLabs.
4. Double-click Policy Controller.
5. Double-click AgentLog.
6. Right-click the Agento.log.0 locked file and select Copy.
7. Within the agentLog folder, right-click in an empty space and select **Paste**.

8. Double-click the **Agent0.log-Copy.0** file to view its contents.

9. Scroll down to view the contents. You can press Ctrl+F to find keywords such as any identifying word from your policy definitions, words common to ABAC activity such as **allow** or **deny**, or words native to NextLabs logging such as **effect** =.

   a. Examples of information found in this **Agent0.log-Copy.0** file:

   i. All of the policies evaluated during one instance of access:
ii. An allow decision was evaluated when this example user, Jorge Gonzalez, logged into the Runabout Air SharePoint:
queryDecisionEngine

INFO: Request 2342972204282387 input params
to

application

pid: 5140

environment

request_id: 2342972204282387
time_since_last_successful_heartbeat: 31

host

inet_address: 184536844

operating-system-user

id: S-1-5-21-972639958-268376111-2639239546-1138

action

name: OPEN

sendto

from
title: relying party inc - root site
cd::id: sharepoint://sharepoint.abac.test/
name: relying party inc - root site
sub_type: site
type: site
ce::destinytype: portal
url: sharepoint://sharepoint.abac.test/
user

id: S-1-5-21-972639958-268376111-2639239546-1138
title: Scientist
department: Research and development
stafflevel: Senior
upn: jgonzalez@ABAC.TEST
company: Conway
name: abac\jgonzalez
clearance: Top Secret

Ignore obligation = false
Process Token = 984
LogLevel = 3
Result: Effect = allow (total:4608ms, setup:4605ms,
obligations:0ms)

Obligations:

From file list: [sharepoint://sharepoint.abac.test/]
To filename list: null
Leveraging NextLabs Control Center Reporter for Reporting and Auditing Purposes

9.1 Introduction ...................................................................................................................... 348
9.2 Introduction to NextLabs Control Center Reporter ........................................................... 349
9.3 Introduction to Reporter Dashboard .................................................................................. 350
9.4 Introduction to Defining and Running Custom Reports in Reporter ................................. 354
9.5 Example Custom Report and Available Formats ............................................................. 361
9.6 Further Example Custom Reports from our Build ........................................................... 370
9.1 Introduction

In previous sections of this How-To Guide (Chapter 7), we installed several NextLabs products that can be used to define and deploy Attribute-Based Access Control policies and enforce decisions regarding user access to Microsoft SharePoint resources based on user, object, environmental attributes, and the corresponding policies in place. We also illustrated how to use and configure the NextLabs Policy Studio, the product responsible for Policy Lifecycle Management, and discussed policy strategy and the translation of business logic into policy (Chapter 8).

In this section of the How-To Guide, we will illustrate how to use the NextLabs Control Center Reporter, a component of the previously installed NextLabs Control Center (Chapter 7), in order to generate reports and provide a graphical user interface for prior policy evaluation and access control decisions in your environment.

Reporter is automatically installed during the NextLabs Control Center installation, which was detailed in chapter 7. In this How-To section we will introduce Reporter, its purpose, interface, and capabilities, then illustrate some example uses based on our build.

9.1.1 Components Used in this How-To Guide

1. NextLabs Control Center Reporter v7.5.0 (64) – web application and graphical user interface for evaluating prior policy evaluation access control decisions and generating reports for monitoring and auditing.

9.1.2 Pre-requisites to Complete Prior to This How-To Guide

1. If you intend to do a setup without identity federation and federated logins, you must:
   a. Install and configure Active Directory (see Chapter 2)
   b. Install and configure Microsoft SharePoint (see Chapter 4)
   c. Install and configure NextLabs Control Center, Policy Studio, and Policy Controller (see Chapter 7)
   d. Define and deploy policies based on your business rules (see Chapter 8)

2. If you intend to incorporate a trust relationship between an IdP and RP and use federated logins into SharePoint, you must:
   a. Install and configure Active Directory (see Chapter 2)
   b. Setup and configure the RP and IdP (see Chapter 3)
   c. Install and configure Microsoft SharePoint (see Chapter 4)
   d. Configure the SharePoint federated login with the RP (see Chapter 5)
   e. Configure the attribute flow between all endpoints (see Chapter 6)
   f. Install and configure NextLabs Control Center, Policy Studio, and Policy Controller (see Chapter 7)
   g. Define and deploy policies based on your business rules (see Chapter 8)
9.2 Introduction to NextLabs Control Center Reporter

The NextLabs Control Center Reporter is a web application that can be used to generate reports on how information is being used in your environment. You can use Reporter to define and run custom queries about policy enforcement activities that are recorded in the Activity Journal, a native, automatic logging mechanism built into the NextLabs SQL database that was configured during installation of the NextLabs Control Center (Chapter 7). These queries are referred to as reports. Reports can be designed to answer a wide variety of questions, such as who has access to certain documents, who is using which resources and when, what types of policy enforcement is taking place, what activity occurred within a given department, and so on.

In addition to reports, you can also use Reporter to create monitors that trigger alerts when specified policy enforcement criteria are met. You can design monitors to cover a wide range of scenarios, such as sending an alert through email when access to a certain resource has been denied more than a specified number of times in a given time period; or when the volume of classified documents that have been downloaded in a given time period exceeds a specific file size. Together, monitors and alerts can provide continuous coverage of critical policy enforcements in an enterprise, as well as a notification system that lets you know when action is required.

User permissions are defined in the Administrator application (another component of Control Center installed in Chapter 7), by creating a new User and assigning one of the four available roles to it. By default, all roles include permission to open and use the reporting functionality of Reporter.

9.2.1 Opening Reporter

1. On the server where NextLabs Control Center was installed, open a web browser (i.e., SQL Server in this build).

2. Enter the URL and press Enter: https://<hostname>/reporter, i.e., https://localhost/reporter
3. At the Reporter login screen, enter valid credentials, such as the Control Center Administrator account created in chapter 7. Click Login.

4. In your browser, the Reporter opening view defaults to the Dashboard tab. The Dashboard tab, Reports tab, and Monitoring tab will be discussed more thoroughly in subsequent sections of this How-To Guide.

9.3 Introduction to Reporter Dashboard

The Reporter Dashboard is divided into panes, each displaying a predefined statistical view of data that provides a snapshot of policy enforcement trends. In the default configuration of Reporter, these panes display data in the following graphs (from the NextLabs Control Center Reporter User Guide, available only to customers at this time):
<table>
<thead>
<tr>
<th>Graph</th>
<th>Description</th>
<th>May Indicate:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top Five Deny Policies</strong></td>
<td>Pie chart representing the five Deny policies that were most frequently enforced over the previous thirty days.</td>
<td></td>
</tr>
<tr>
<td>(Month)</td>
<td></td>
<td>• Misunderstanding of access level: users being blocked from a resource they believe they should use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Incorrectly defined entitlements: users should have access, but policies are not updated or correctly designed</td>
</tr>
<tr>
<td><strong>Top Ten Denied Users</strong></td>
<td>Bar chart representing the ten users who have had the most instances of any Deny policy enforced against them.</td>
<td></td>
</tr>
<tr>
<td>(Month)</td>
<td></td>
<td>• Users who habitually snoop into resources they are not authorized to use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Incorrectly defined entitlements: users or group should have access, but policies are not updated or are incorrectly designed</td>
</tr>
<tr>
<td><strong>Top Five Deny Resources</strong></td>
<td>Bar chart representing the five resources that any users have most frequently attempted to access and been blocked by an active policy, over the previous seven days.</td>
<td></td>
</tr>
<tr>
<td>(Week)</td>
<td></td>
<td>• Resources of broad interest to users who should not be using them</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Incorrectly designed resource or user component, blocking users who should have access</td>
</tr>
<tr>
<td><strong>Top Five Allow Resources</strong></td>
<td>Bar chart representing the five resources that users have most frequently attempted to access and been allowed by an active policy, over the previous seven days.</td>
<td></td>
</tr>
<tr>
<td>(Week)</td>
<td></td>
<td>• Improperly designed resource component or policies, which allow inappropriate users access to sensitive resources</td>
</tr>
<tr>
<td><strong>Deny Policy Enforcement Trends</strong></td>
<td>Bar chart representing the trend, over the previous 30 days, of the daily total instances of any deny policy being enforced on any user, for any resource.</td>
<td></td>
</tr>
<tr>
<td>(Month)</td>
<td></td>
<td>• Progress (or lack thereof) in educating users about access policies and individual/group entitlements, at a broad level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Improperly designed policies that are blocking too many users who expect and are entitled to access or use</td>
</tr>
<tr>
<td><strong>Recent Allows</strong></td>
<td>List of details about the most recent ten instances of any allow policy being enforced against any user, for any resource. Details listed include:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Instances where some urgent action is required, such as users being allowed access to some resource they should not be using, due to lack of policy coverage or an incorrectly defined policy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Date of enforcement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Name of enforced policy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• User who triggered the policy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Action that triggered the policy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Resource the user was trying to access</td>
</tr>
</tbody>
</table>
These panels are configurable such that an administrator can choose which panels and data are visible and how they are laid out within the Dashboard according to the business’s business logic, policies, and priorities.

The data displayed in all panes of the dashboard is refreshed from the Activity Journal each time you open the Dashboard tab. This means that data is updated on demand; for example, if a pane shows some statistic for the past week, that reflects not the last seven whole calendar days, but the last seven 24-hour periods starting from the top of the current hour.

### 9.3.1 Exploring the Dashboard

1. On the server where NextLabs Control Center was installed, open a web browser, i.e., SQL Server in this build
2. Enter the URL and press Enter: https://<hostname>/reporter, i.e., https://localhost/reporter
3. At the Reporter login screen, enter valid credentials such as the Control Center Administrator account created in chapter 7. Click Login.

4. In your browser, the Reporter will default to the Dashboard tab.

5. The charts and graphs on the Dashboard are interactive. When you move your cursor over a bar in a bar chart or a slice in the pie chart, a tooltip displays information about that value series.

6. Example seen in the image below: 36.4% of the Deny policies evaluated in the last 30 days belonged to the SharePoint Protection – Department policy set.
7. Another example from this build seen in the image below: in the Deny Policies trend in the last 30 days, June 26, 2015 saw an unusually large number of Deny Policies relative to other days.

#### 9.4 Introduction to Defining and Running Custom Reports in Reporter

In Reporter, you can define and run reports in the Reports tab. This tab is divided into two panes, **Saved Reports** on the left side of the Reports tab window and **Report Details** on the right.
The Saved Reports pane provides a list of all saved reports available to you. This includes all reports you create and save, all reports saved by other users and marked as Shared, and the sample reports used to generate data that is displayed in the Dashboard tab. When you click on any item in Saved Reports, the details of that report are displayed in Report Details on the right. This is also where you work when you create a new report.

In the Report Details pane, define the following:

- The time period of the policy activity data to cover in the report
- The criteria, or filters, that determine what policy activity data to include in the report
- The output format of the report

The default settings in Report Details display when you click the Reports tab or when you click New in the Saved Reports pane. By default, the time period for the report is the current day, all policy activity data at the user level is included, and the data is presented in table format.

After defining a new report or editing an existing report, click Run at the bottom of the Report Details pane to view the results, which we will illustrate in the following two subsections.

### 9.4.1 Defining a Custom Report

In this subsection we will list the standard steps for creating a custom report. In section 9.5 of this How-To Guide we will illustrate some example custom report sections that demonstrate Reporter’s report capabilities.
9.4.1.1 Logging into Reporter

Before being able to define a custom report, you must first log in to Reporter and click on the Reports tab as seen in the steps below:

1. On the server where NextLabs Control Center was installed in chapter 7, open a web browser, i.e., SQL Server in this build.

2. Enter the URL and press Enter: https://<hostname>/reporter, i.e., https://localhost/reporter

3. At the Reporter login screen, enter valid credentials, such as the Control Center Administrator account created in chapter 7. Click Login.

4. In your browser, the Reporter user interface will default to the Dashboard tab. The Dashboard tab, Reports tab, and Monitoring tab will be discussed more thoroughly in subsequent sections of this How-To Guide.
5. Click on the **Reports** tab to open the Reports tab window.

### 9.4.1.2 Defining the Custom Report

In order to define a custom or new report, you must specify filters and change default settings within the Report Details – Report Query pane. If you don’t specify any filters or change any of the default settings, the report retrieves all policy activity data categorized as user-level events for the current day.
1. In the Report Details – Report Query pane, define the report query by filling in data or using drop-down menus to define your desired report.
   a. Note: Many of the fields are optional. Required fields contain default values.
      i. In the From and To fields, specify the start date and time, and end date and time, respectively, of the time period you want the report to cover. Click in the field to choose a date and time from the calendar. When specifying a report period, be sure to consider the time zone where Control Center is installed, and the time period of data stored in the Activity Journal.
      i. In Event Level, select the level of event verbosity the report contains:
         ° User Events (default): Logged in the Activity Journal as Level 1
         ° Application Events (application and user-level events): Logged in the Activity Journal as Level 2
         ° All System Events (system, application, and user-level events): Logged in the Activity Journal as Level 3
         ° Note: As a rule, you should leave this setting at User Events. This setting significantly reduces the amount of system noise. Application- or system-level events generally are not useful in monitoring policy or user activities.
   2. In Decision, select the type of enforcement effect to include in this report:
      a. Allow: Instances when the policy permitted the user to perform the action covered by the policy. Note that the report results always depend on what information is logged. If
the policy does not have any On Allow logging obligation specified, this report will not return any On Allow data whether or not you select this option.

b. Deny: Instances when the policy did not allow the user to perform the action. Deny decisions are always logged.

c. Both: All instances when the policy was enforced, with either Allow or Deny effect.

3. In Action, select the user action or actions to include in this report. The list shows all currently defined actions.

a. To select multiple actions, hold Ctrl and click each action. If you do not make any selections, all actions are included.

b. Note: Policies involving Paste actions do not support logging obligations, therefore, instances of their enforcement are not included in reports.

4. In User, specify one or more users on which to filter the activity data, or leave this field blank to include all users. Use the User Lookup window (magnifying glass icon) to browse through all users currently defined in your Information Network Directory, and select the users you want.

5. In User Criteria, specify additional user criteria by creating one or more conditions. Each condition consists of a user attribute, an operator, and a value. You must click the + button to add a condition to the query.

6. In Resource Path, type the network path of the resource on which to filter, or leave this field blank to include all resources.

7. In Resource Criteria, specify additional resource criteria by creating one or more conditions. Each condition consists of a resource attribute, an operator, and a value. Click the + button to add a condition to the query.

8. In Policy Name, specify one or more policies on which to filter, or leave this field blank to include all policies. Use the Policy Lookup window to browse through and select which policies you want to include.

9. In Policy Criteria, specify additional policy criteria by creating one or more conditions. Each condition consists of a policy attribute, an operator, and a value. Click the + button to add a condition to the query.

10. In Other Criteria, specify additional criteria by creating one or more conditions. Each condition consists of a general attribute (for example, host name, host IP, and application name), an operator, and a value. Click the + button to add a condition to the query.
9.4.1.3 Setting the Custom Report Display Options

Within the Report Details – Report Query pane, directly below the Other Criteria filter, continue with these steps to set the display options for your custom report:

1. In **Report Type**, select the output format in which to display the data: Table, Bar Chart, Horizontal Bar Chart, or Pie Chart. Use a table to display policy activity details in a row-and-column format. Use a chart to display a summary of policy activities.

2. If you selected one of the charts in Report Type, in **Show**, select a grouping option. Grouping is not available to a table.
   a. Group by User: The chart shows the number of enforcement events for each user covered by the report.
   b. Group by Resource: The chart shows the number of enforcement events for each resource covered by the report.
   c. Group by Policy: The chart shows the number of enforcement events for each policy covered by the report.
   d. Group by Month: The chart shows the number of enforcement events for each month covered by the report. Select this option only if the time period you specified spans more than one month.
   e. Group by Day: The chart shows the number of enforcement events for each day covered by the report.

3. In **Sort By**, select a field on which to sort the data, then select Asc to sort in ascending order or Desc to sort in descending order. If the report is a table, you can sort the data by any attribute. If the report is a chart, you can sort either by the grouping item (user, resource, policy, month, or day) or by Result Count (the number of enforcement events for each user, resource, policy, month, or day).

4. In **Max Results**, specify the maximum number of results to display in the table or chart. For charts, this number represents the maximum number of bars in a bar chart, or slices in a pie chart. For readability reasons, charts should display a limited number of bars or slices. For a table, the number represents the maximum number of rows (each row represents an event). Tables that show a large number of rows present the data on multiple pages.

5. In **Display Columns**, select the columns to display in a table. This setting applies to tables only. USER_NAME, POLICY_FULLNAME, POLICY_DECISION, HOST_NAME, and APPLICATION_NAME are selected by default. To remove any of those columns or to add other columns, click and use the arrow icons to move columns out of, or into, the Selected pane.
9.4.2 Running a Custom Report

Directly beneath the filters and data fields for defining the report and setting its display settings, do the following in order to run the report and/or save it for the future:

1. At the bottom of the Report Details – Report Query pane, click Run to generate the new report.

2. If you want to run this report again in the future, save the report. Click Options, and select Save.

9.5 Example Custom Report and Available Formats

In this section we will present examples of different report formats, all representing a small set of event data, returned by the same custom report from our build. By comparing the example formats, you will gain a better understanding of the way the different formats can be used to highlight different aspects of the same data depending on your business rules or priorities.

The custom report used in this section will result from a query that requests all events by users on all resources for one week (June 7, 2015 to June 13, 2015). We include columns that are relevant for our example business logic and the ABAC policies we put in place in chapter 8. For example, we chose to include the Department and Sensitivity columns, which were custom attributes in the metadata we added to the documents uploaded to the RP’s SharePoint sites.

9.5.1 Defining the Example Custom Report

9.5.1.1 Customizing Report Query Fields for this Report

1. In the Report Query pane, change the fields for the From and To date to match the desired query for the week of June 7, 2015 to June 13, 2015.

2. In the Report Query pane, click on the Max Results field to open the drop-down menu. We chose 11 for demonstration purposes.
3. In the Report Query pane, leave the rest of the fields in the default query settings.

9.5.1.2 Editing the Columns for Custom Views

1. Toward the bottom of the Report Query pane, click on the columns icon at the end of the Display Columns line of text to open the Select Display Column window.
2. In the Select Display Column window, in the **Available** attribute list, review standard attributes (i.e. Action, Log_Level, Host_IP, etc) and custom attributes (department, sensitivity).

![Select Display Column window](image)

3. Click on any available attribute of interest to highlight it, then click the single right arrow button to add it to the list of **Selected** attributes.

4. The attribute name will move from the **Available** list to the **Selected** list.

5. **Note**: Attributes can be added and removed individually by using the single arrow buttons between lists, or as a group by using the double arrow buttons between lists.

![Select Display Column window](image)

9.5.1.3 Running the Report Query

6. At the bottom of the Report Query pane, click **Run** to run the query. (**Tip**: You can click on **Options** and **Save** or **Save As** to save the query for future use.)

![Run button](image)

7. Scroll down in your browser window to see the Results pane illustrated in the following section.
9.5.2 Format: Table of Event Data

The default results pane with the display columns you selected displays showing the query results. This is illustrated in the following image.

This excerpt from the query results shows that:

- 13 pages of policy enforcement events were logged.
- All events in this excerpt occurred on June 12, 2015 (as illustrated in the Date column).
- Each event from this excerpt was triggered by the same user, who had logged in with a federated identity from the IdP (chapters 1 through 5).
- Each event corresponds to one of three policies: SharePoint Protection – Sensitivity, SharePoint Protection – Maintenance Denied 5am-5pm, or SharePoint Protection – Department.
- Five resources were involved:
  - The first row shows that the resource was an .rtf document from the Internet Technology department’s SharePoint sub-site, marked at sensitivity level 3.
  - The second through fourth rows show that the resource was the Internet Technology department site.
  - The fifth through seventh rows show that the resources were the underlying .css style sheet and logo used on the SharePoint site.
  - The seventh through tenth rows (up to the second to last) show that the resources were the underlying .css style sheet and logo used on the SharePoint site.
  - The eleventh and final row from this excerpt shows that the resource was another .rtf document from the Internet Technology department SharePoint sub-site, marked at sensitivity level 1.
- In the case of three out of the five resources, the enforcement decision was Allow, as shown in the fourth column (second through tenth rows).
- In the case of two out of the five resources, the enforcement decision was Deny, as shown in the fourth column (first and last rows).

Keep these details in mind as you analyze the data in the following charts.
9.5.3 Format: Bar Chart Grouped by Policy Chart

Grouping events by policy is useful for identifying policies that are being triggered with unexpected frequency, which may be an indication that they are improperly designed and cover users, resources or actions that they should not. It can also indicate concentrated efforts at unauthorized data access. To examine the latter possibility, it is often helpful to switch to the Group by User option in order to focus on who is performing the activity, as seen in section 9.5.4.

9.5.3.1 Customizing the Display Settings

1. Using the Report Details – Report Query window from section 9.5.2 for displaying the results in Table format, make the following edits to display results in a Bar Chart grouped by Policy:
   a. From the Report Type list, select Bar Chart.
   b. From the Show list, select Group by Policy.
   c. From the Sort By list, select Policy.
   d. From the Max Results list, choose a number or type one in the field.
      Example: The value 6 means that our bar chart will display up to six policies, including but not limited to the number of policies displayed in the Table format.
   e. Click on the Asc (Ascending) radio button to set the sorting order.

9.5.3.2 Running the Report Query

1. At the bottom of the Report Query pane, click Run to run the query

9.5.3.3 Viewing the Results as a Bar Chart Grouped by Policy

1. In the same browser window, scroll down if necessary. Under the Run button, review the resulting Bar Chart Grouped by Policy.

As illustrated below, hundreds of enforcement decisions were logged during the week, and the three most commonly evaluated policies include two that were included in the table from section 9.5.2, formatting results by Table.
When the same data is grouped by user, and the bar chart is selected, the following chart is generated. As noted previously, the four policies were each triggered by a different user, so the graph shows four bars—each representing one user. Each is labeled with a user name. In this example, the bars are the same height, since each of the four users triggered a policy once.
Chapter 9. Leveraging NextLabs Control Center Reporter for Reporting and Auditing Purposes

9.5.4.1 Customizing the display settings

1. Using the same Report Details – Report Query window from the previous subsection, make the following edits to display results in a Bar Chart Grouped by Policy.
   
a. From the Report Type list, select Bar Chart.
   
b. From the Show list, select Group by User.
   
c. From the Sort By list, select User.
   
d. From the Max Results list, choose a number or type one in the field.
      
      Example: The value 6 indicates that this will be the maximum number of users reflected in our Bar Chart.
   
e. Leave Asc selected.

9.5.4.2 Running the Report Query

1. At the bottom of the Report Query pane, click Run to run the query.

9.5.4.3 Viewing the Results as a Bar Chart Grouped by User

1. In the same browser window, scroll down if necessary. Under the Run button, review the resulting Bar Chart Grouped by User:

As illustrated below, only five users were accessing the protected RP SharePoint resources during this week period, and all logged in via federated identity from the IdP:

- Two users had very minimal activity logged during this week: schen@abac.test and sharepointadmin@abac.test
- Two users had relatively similar activity logged during this week: jdoe@abac.test and jgonzalez@abac.test
- One user had an extremely large amount of activity logged during this week: smith@abac.test
9.5.5 Format: Pie Chart Grouped by Resource

The Group by Resource option shows the extent of specified events—in this case, policies being triggered—per individual resource covered by the report.
Because policies often cover large numbers of individual documents or other resources, grouping by resource is only helpful when the number of events has already been narrowed down to a smaller set by various report filters, such as policies or users. A pie chart is ideal here, because in the context of resource use, the relative access activity regarding some single file or other resource as compared to all others is generally of more interest than any absolute number of instances of access.

### 9.5.5.1 Customizing the Display Settings

1. Using the same Report Details – Report Query window from the previous subsection, make the following edits to display results in a Bar Chart grouped by Policy
   a. From the **Report Type** list, select **Pie Chart**.
   b. From the **Show** list, select **Group by Resource**.
   c. From the **Sort By** list, select **Resource**.
   d. From the **Max Results** list, select a number or type one.
      Example: The value 10 means that will be the maximum number of resources displayed in our Pie Chart.
   e. Leave **Asc** selected.

### 9.5.5.2 Running the Report Query

1. At the bottom of the Report Query pane, click **Run** to run the query.

### 9.5.5.3 Viewing the Results as a Bar Chart Grouped by User

1. In the same browser window, scroll down if necessary. Under the **Run** button, review the resulting Bar Chart Grouped by Policy:

   As illustrated below, the maximum of ten resources are displayed in the pie chart.

   - The most commonly accessed resource during this week period (69.5%) was our build’s SharePoint home page.
   - The two second-most accessed resources during this week period were the ABAC IT department and its forms sub-site (where documents are stored).
• The remaining seven most-accessed resources during this week after the top three have relatively very minimal access, and the majority of those are documents that belong to specific department sub-sites, such as Finance Dept Quarterly Reports, IT Dept System Configuration documents, etc.

9.6 Further Example Custom Reports from our Build

In this section we will illustrate how to define custom reports that will provide a graphical representation of particular kinds of activity that could be of interest to our RP business.

For our first additional example we will use a fictitious user from our build’s IdP and check her activity on the RP SharePoint site within a specific time period. The report we define will focus on the user Lucy Smith (username: lsmith) and all of her Allowed and Denied access during a specific timeframe, such as May 1, 2015 – June 30, 2015.

For our second additional example we will use a document on the RP SharePoint site that has been marked with a metadata attribute called sensitivity. The document’s sensitivity value is set to 3, which according to our example ABAC policies requires that 1) the user accessing the document belongs to the same or appropriate department for accessing it, 2) the access occurs during regular business hours Monday-Friday, and 3) the user has a clearance attribute value of Top Secret. The report we define will focus on the access attempts on that document for the months of May and June 2015.

9.6.1 Custom Report Illustrating One User’s Access During Two Months

1. Follow the steps for section 9.5.4, Format: Bar Chart Grouped by User, and change the From field to May 1, 2015 and the To field to June 30, 2015.

2. Within the browser, in the results area at the bottom of the Report Details window, click on the vertical bar that represents the user smith@abac.test or abac\lsmith (light green, the far-right bar in our chart below).

The Report window of your browser will automatically refresh, and a default query on the User will run automatically.
3. Within the browser window, scroll up to Report Details and verify that the User: field was automatically populated with abac\{smith}.

In the Report Query pane you will see that the default query pertaining to the User has a Report type of Table, sorted by date in descending order, with a maximum of 100 results.
4. Within the browser window, scroll back down to the resulting Table to review its data. See the excerpt below.

If desired, you can change the Display Columns, Report Type, etc. to customize your view as illustrated in previous subsections.

9.6.2 Viewing Access Attempts on Individual Resources

This section provides instructions for creating a custom report that shows the access attempts of a single resource for a period of two months.

1. Follow the steps for section 9.5.5, Format: Pie Chart Grouped by Resource, and change the From field to May 1, 2015 and the To field to June 30, 2015.

2. From the resulting list of resources under the pie chart, find the color of a resource with a name including level 3, which according to our schema means in SharePoint metadata the sensitivity level attribute is equal to 3.
3. Click on that resource in the pie chart (example: light pink area of 2.3% is for a Sales Dept document called *sales document 2015 – level 3.txt*).

This will begin an automatic default query for that resource similar to the one done above based on the user lsmith.

4. Within the browser window, scroll up to Report Details and verify that the Resource Name: field was automatically populated with the name *Sales document 2015 – level 3.txt*.

In the Report Query pane, you will see that the default query pertaining to the resource has a Report type of Table, sorted by date in descending order, with a maximum of 100 results.
5. Within the browser window, scroll back down to the resulting table to review its data. See the excerpt below.

If desired, you can change the Display Columns, Report Type, etc. to customize your view as illustrated in previous subsections.
10 Configuring a Secondary Attribute Provider

10.1 Introduction ...................................................................................................................... 376
10.2 Component Software and Hardware Requirements ........................................................ 381
10.3 Ping Custom Data Store .................................................................................................. 382
10.4 NextLabs PIP Plugin ........................................................................................................ 388
10.5 Protocol Broker .............................................................................................................. 393
10.6 Apache Directory Service (ApacheDS) ............................................................................ 402
10.7 PingFederate - Apache Integration .................................................................................. 409
10.8 Configuration of PingFederate to Query the JIT Cache when Responding to Secondary Attribute Requests 425
10.9 ApacheDS Schema Extension ......................................................................................... 445
10.10 Functional Tests .......................................................................................................... 459
10.1 Introduction

This chapter provides a description of the architecture, compilation, and deployment instructions for a secondary attribute provider and its components, which we describe as a custom Policy information point (PIP), to be included as part of the ABAC infrastructure. We also demonstrate how to configure the Relying Party server to accommodate the custom PIP and its component JIT provisioning mechanism.

The secondary attribute provider comes into the picture when a user tries to access a resource at the Relying Party's Resource Provider, and the Policy decision point (PDP) finds that an essential attribute needed to make the access control decision is missing from the initial set of attributes sent from the Identity Provider. In our build, this would mean a user with a federated identity (via PingFederate Identity Provider, IdP, augmented with two-factor authentication by RSA AA) has already logged into Microsoft SharePoint (Relying Party's Resource Provider), but when trying to open a particular resource on the site, the NextLabs Policy Controller (PDP) makes a run-time decision that additional subject attributes are needed before the access decision can be made. The PDP determines this while evaluating the existing ABAC policies (created in the NextLabs Policy Studio, PAP in our ABAC build) against the user, resource, and environmental attributes at play at the time of requested access.

Providing the secondary attribute collection capability in our build required the implementation of new components and related features, which we will describe more in detail later in the chapter:

- NextLabs Policy Information Point (PIP) Plugin to extend the NextLabs Policy Controller (PDP) when additional attribute(s) are needed
- Protocol broker to initiate and receive a SAML attribute query and SAML response
- Custom data store plugin for PingFederate on the Relying Party (RP) server which will cache attributes in order to limit the number of secondary requests to the PingFederate Identity Provider (IdP) server
- Apache Directory Server (ApacheDS), an LDAP in which PingFederate can create and update local user accounts and associated attributes based on the attributes contained in SAML assertions received after authentication from IdP
- PingFederate RP configuration must be modified so that it can serve as an IdP as needed, such as when checking its JIT cache (Apache DS LDAP) before sending requests to the IdP

In later sub-sections of this chapter we will discuss in detail the purpose of each of these new components and features, and how they are developed, configured, compiled, and deployed.

Note: The custom PIP we have developed involves new custom components, open source components, and commercially available components. For open source and commercial components, the related descriptions in this chapter have been limited to installation and relevant configuration required for the desired functionality of our build. If you are interested in other details or additional capabilities of this software, explore the referenced product literature or contact that organization.
10.1.1 Prerequisites

In order to follow the instructions of this chapter, it is necessary that seven of the previous How-To sections have been successfully completed. The required components that must be installed and configured before continuing in this chapter include:

1. Installation and Configuration of Active Directory (Chapter 2)
2. Installation and Configuration of RSA AA (Chapter 2)
3. Installation and Configuration of RSA AA Plugin (Chapter 2)
4. Installation and Configuration of PingFederate on both the RP and IdP federation servers (Chapter 2 and Chapter 3),
5. Installation and Configuration of Microsoft SharePoint (Chapter 4 and Chapter 5)
6. Configuration of the attribute flow (Chapter 6)
7. Installation and Configuration of NextLabs Control Center, Policy Studio, Policy Controller, and Entitlement Manager for SharePoint Server (Chapter 7)

10.1.2 Criteria for Secondary Attribute Collection

At the time of ABAC policy evaluation, required attributes may not be available or the system may not find it appropriate to use for various reasons, including, but not limited to:

- For security and privacy purposes it is not ideal to acquire all known attributes for a subject when the session is created. Some attributes may be PII or of higher sensitivity and should not be sent to the Relying Party until an access request made by the user requires those attributes.
- Depending on the longevity of a session, attributes risk becoming stale. Because of this potential for staleness, it is essential to procure attributes as needed, depending on the freshness criteria established by the system. The freshness of attributes is sometimes guided by the policies established for a local cache.
- The attribute needed for a specific attribute request may not be attributed owned by the identity provider but rather may need to be acquired from an external party attribute provider.

10.1.3 Components

The custom PIP described in this chapter is composed of four new components and mechanisms which interact or integrate with different existing components in our ABAC build as extensions, plugins, or web applications:

- **NextLabs Plugin**: This plugin extends the NextLabs Policy Controller to make attributes available based on the criteria mentioned in section 10.1.2, when the PDP determines that attribute values needed to evaluate an ABAC policy are insufficient or unavailable. Following the recommendation in the software development framework provided by NextLabs, the NCCoE implemented this PIP plugin in Java, and deployed the plugin within the NextLabs Policy Controller software architecture on the server we call SharePoint server
in our build. Due to the requirements of the Policy Controller architecture, the plugin can request the values of multiple missing attributes sequentially, one at a time.

- **Protocol Broker**: This agent, in the form of servlet local to the NextLabs installation, is responsible for facilitating communication between the NextLabs PIP Plugin and the PingFederate RP server following an Assertion Query/Request SAML2 Profile. This web application is deployed on a tomcat server that listens on localhost(127.0.0.1) and only communicates using https with mutual TLS. Similar to the NextLabs PIP Plugin, this component is also installed on the SharePoint server.

- **Ping Custom Data store**: This custom data store is an extension built using Ping SDK. It enables the RP server to query the IdP server and coordinates resulting attribute values back to the RP. When it is chained with a built-in data store to query JIT Cache (LDAP), it enables RP to provide data from and configuration to various data stores (JIT in this build). This helps the custom data store to query and coordinate the result from local JIT and remote Active Directory at the PingFederate IdP.

**Just-in-Time provisioning** is a feature provided by PingFederate to store attributes of a subject for a limited time. We implemented JIT provisioning using ApacheDS. ApacheDS 2.0 is an embeddable, extendable, standards compliant, modern LDAP server written entirely in Java, and available under the Apache Software License. It also supports network protocols like Kerberos and NTP. PingFederate RP acts as an IdP for the secondary attribute provider. To fulfill in this role, the PingFederate administrative console provides mechanisms to configure SP and IdP connections. These configurations manage connection settings to support the exchange of federation-protocol messages. It also allows configuration of data stores within the connection and an attribute contract that acts as the medium to convey attribute mapping from one entity to another.
10.1.3.1 Sequence Diagram of Custom PIP Component Interactions

Figure 10.1 Architecture
**Description**

Nextlabs PDP (Policy Controller) is the arbitrator for all access decisions at the SharePoint portal. It controls access to SharePoint URL(s) by evaluating rules against the attributes of the entities (subject and object), actions, and the environment relevant to a request. It may be possible that the attribute required for the decision is not available at run time. In that case, it looks for the registered plugin that will fetch the attribute using the following flow:

1. When the policy controller does not receive the attributes required to make a decision, a secondary attribute request will be initiated by calling the PIP Plugin.

2. PIP Plugin is a registered plugin with the NextLabs Policy Controller. It implements the interface dictated by the NextLabs software. By virtue of this implementation, it receives the subject and name of the attribute that is required for the policy decision.

3. When the subject and attribute name are received, the PIP Plugin checks its local short-term cache (in this build, configured to hold values for two seconds) to see if the needed attribute for the subject was recently requested.

4. If the attribute is still in cache, the value is returned to the Policy Controller. If the value is not in cache, the PIP Plugin initiates an HTTPS request to the Protocol Broker.

5. The Protocol Broker receives the attribute name and subject from the HTTPS request and forwards them as a signed SAML 2.0 Attribute Query to PingFederate-RP on a channel protected by mutual TLS.

6. Once PingFederate-RP receives the SAML 2.0 attribute query, it sends an LDAP request to the JIT cache to see if the attribute was previously queried in a secondary request.

7. If the subject does not have the attribute value assigned in the JIT cache, PingFederate-RP will forward the subject and attribute name to the Custom Data Store plugin. The Custom Data Store plugin acts as a pointer back to the PingFederate-IdP. To do this, the Custom Data Store dispatches an HTTPS request to the PingFederate-RP with the PingFederate-IdP as the attribute query point.

8. PingFederate uses an HTTPS query to form a SAML 2.0 attribute query and dispatch it to the PingFederate at the IdP.

9. The PingFederate at the IdP accepts the SAML 2.0 request, verifies if the user has the attribute of need, and replies back to the PingFederate-RP with a SAML 2.0 response.

10. PingFederate-RP validates the SAML 2.0 response, retrieves attribute values, and responds to the original Custom Data Store HTTP request with the attribute values.

11. The Custom Data Store then responds to the PingFederate-RP attribute request with an attribute response.

12. The PingFederate-RP constructs a SAML 2.0 response and sends it to the Protocol Broker.

13. The Protocol Broker retrieves the attribute or exception from the SAML 2.0 response and forwards it to the NextLabs plugin, which passes the attribute or exception back to the Policy Controller.
## 10.2 Component Software and Hardware Requirements

<table>
<thead>
<tr>
<th>Component</th>
<th>Server where component is installed</th>
<th>Compilation method</th>
<th>Required software or hardware</th>
<th>Operating System</th>
<th>Optional software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ping Custom Data Store</td>
<td>PingFederate RP server</td>
<td>Ant 1.9.2</td>
<td>PingFederate 7.3.2; Java version same as PingFederate installed</td>
<td>Windows Server 2012</td>
<td></td>
</tr>
<tr>
<td>NextLabs Plugin</td>
<td>SharePoint server</td>
<td>Apache Maven 3.2.5</td>
<td>SharePoint 2013; NextLabs Entitlement Manager for SharePoint Server, NextLabs Policy Controller, NextLabs Control Center, NextLabs Policy Studio; SQL Server 2012; Java version same as NextLabs Policy Controller installed (1.6)</td>
<td>Windows Server 2012</td>
<td>BareTail (used here as a log file annotator)C Copyright Bare Metal Software Pty Ltd. Download 05/22/2015.</td>
</tr>
<tr>
<td>Protocol Broker</td>
<td>SharePoint server</td>
<td>Apache Maven 3.2.5</td>
<td>PingFederate 7.3.2; SharePoint 2013; NextLabs Entitlement Manager for SharePoint Server, NextLabs Policy Controller, NextLabs Control Center, NextLabs Policy Studio; SQL Server 2012; Java version same as NextLabs Policy Controller installed (1.6)</td>
<td>Windows Server 2012</td>
<td></td>
</tr>
<tr>
<td>Apache Directory Server</td>
<td>N/A</td>
<td></td>
<td>PingFederate 7.3.2; Java 7.0 (recommended by Oracle's JDK. Some issues have been reported with Java 8); 384 MB of memory by default, can be changed using Apache Directory Studio (included)</td>
<td>Windows Server 2012</td>
<td></td>
</tr>
</tbody>
</table>
10.3 Ping Custom Data Store

10.3.1 Functionality and Architecture

This data store was developed according to the guidelines from the Ping Identity provided [here](#). It has three functionalities:

- **Configuration**
  - HttpConfig class is used to read in a configuration file for the custom data store. Configuration parameters, like truststore location, password and attribute names can be defined in a file and read in as a configuration by HttpConfig class. The structure of the HttpConfig class configuration is based on spring annotation.
  - Other sets of configuration can be read via a web interface. A detailed description of these parameters is provided in step 9 of section 10.3.4 in this How-To guide.

- **Communication**
  - Similarly, dispatching the http request relies on PingClient class. PingClient uses classes under the spring http package. PingClient sends an https query to Attribute Query End Point. All of the parameters for the https URL are provided by the web interface.

- **Custom Data Store**
  - CustomDataStore is a class that implements com.pingidentity.sources.CustomDataSourceDriver.
  - It implements all methods specified by the contract, i.e.:
    - boolean testConnection(): This method tests whether a host and port is reachable or not. It is assumed that if host and port is reachable, a URL will be available.
    - java.util.List<java.lang.String> getAvailableFields():

The Class Structure and their interactions are provided in the Interaction Diagram and Class Diagram.

---

**Figure 10.2 Ping Custom Data Store Interaction Diagram**
10.3.2 Deploying the Ping Custom Data Store

Note: PingFederate administrator’s manual provides detailed steps for every platform. In our build, we used the Windows Server 2012 platform.

1. Log on to the PingFederate RP server.
2. Click on the Windows icon and begin typing Services.
3. Double-click the Services application icon.
4. Click on the Name column to sort by alphabetical order, and look for PingFederateService.
5. If the status column reads running, right-click on PingFederateService and click Stop.
6. Prepare environment based on PingFederate documentation. This may involve going to ../pingfederate-7.3.0/pingfederate/sdk folder
7. Click on the Windows icon and begin typing Cmd.
8. Double-click the icon to open the Command Prompt.
9. In Command Prompt, navigate to your installation of PingFederate and its sdk folder by typing the following command and pressing Enter. Example: cd C:/pingfederate-7.3.0/pingfederate/sdk/
10. Within the sdk folder, locate build.local.properties and open it with your default text editor. For example, enter the following command and press Enter: notepad build.local.properties
11. In your default text editor (Notepad in our example), set or update target-plugin.name to idp-query-data-store, i.e.,

```
httpConfig

<httpConfig/>

HTTPPostAttributeBuilder

<HTTPPostAttributeBuilder/>

CustomDataStoreConfiguration

<CustomDataStoreConfiguration/>

DefaultTrustManager

<DefaultTrustManager/>

Figura 10.3 Ping Custom Data Store Class Diagram

```

```
# Please set the 'target-plugin.name' property to the name of the directory (under plugin-src) that
# contains the source code of the plugin you want to build.

target-plugin.name=idp-query-data-store

12. Within the Command Prompt window, navigate to your idp-query-data-store folder by enter-
ing a cd command with a path to your idp_query_data_store and pressing Enter. Example: cd c:\--path-to-your-idp_query_data_store

13. Within the Command Prompt window, copy idp-query-data-store along with all subfolders to your PingFederate installation’s sdk/plugin-src folder by entering a cp command and pressing Enter. Example: cp -rf idp_query_data_store C:/pingfederate-7.3.0/pingfederate/sdk/plugin-src

14. Within the Command Prompt window, run the following command and press enter in order to make sure all relevant subfolders exist: ls -ltr ./idp-query-data-store/

- Example results from the above command:

  total 4
  drwxrw-r--. 3 t... t.... 16 Apr 29 11:34 java
  drwxrw-r--. 2 t... t.... 4096 Apr 29 12:59 lib
  drwxrwxr-x. 4 t... t.... 30 May 15 17:52 build
  drwxrw-r--. 2 t... t....51 May 29 09:26 conf

10.3.3 Compilation

The Building and Deploying with Ant chapter of the SDK Developer’s Guide by Ping provides a detailed description of compiling and deploying the project using Apache Ant. For current deployment it may be sufficient.

1. Click on the Windows icon and begin typing the word **Cmd**.

2. Double-click the icon to open the Command Prompt.

3. It is essential to know about the attributes that this data store will return. PingFederate calls the getAvailableFields() method to determine the available fields that could be returned from a query of this data source. These fields are displayed to the PingFederate administrator during the configuration of a data source lookup. The administrator can then select the attributes from the data source and map them to the adapter or attribute contract. PingFederate requires at least one field returned from this method.

4. To change it, go to your ping installation directory. From that directory, navigate to ..\pingfederate-7.3.0\pingfederate\sdk\plugin-src\idp-query-data-store\conf . Open .\config.properties with your favorite editor. Change the value for the attribute called **NameOfAttributes**: 
Chapter 10. Configuring a Secondary Attribute Provider

NameOfAttributes=fullname,username,stafflevel,role,division,employee,c clearance

Use a comma to separate attribute names. More attributes can be added by adding subsequent commas and attribute names.

5. Navigate to your PingFederate sdk folder, i.e., cd C:/pingfederate-7.3.0/pingfederate/sdk/

6. Within the Command prompt window, type the following compilation command and press Enter: ant deploy-plugin

10.3.4 Configuration within PingFederate Administrative Console

The end of successful execution of ant deploy-plugin signals the installation of the data-store driver. Its configuration is provided in detail by Ping documentation. In summary, it spans the following process:

1. Log on to the Ping RP server.
2. Open an internet browser.
3. Enter the following URL and press Enter: https://localhost:9999/pingfederate/app
4. Enter your PingFederate administrator username and password, then click Login.

5. In the browser window, under the Main menu area, find Server Configuration->System Settings->Data Stores. Double-click on Data Stores.
6. At the bottom of the browser window, click **Add New Data Store**.

7. On the Data Store Type screen, select **Custom** and click **Next**.

8. On the Custom Data Store Type screen, specify **Data Store Instance Name** and **Data Store Type**. The name can be arbitrary, but you must select **IDP Attribute Query** from the **Data Store Type** drop-down. Click **Next**.
9. To configure the data store, the following parameters must be configured. These parameters are guided by the requirements of the end point (/sp/startAttributeQuery.ping) defined by Ping documentation here:

https://10.33.7.5:9031/sp/startAttributeQuery.ping?AppId=appid&SharedSecret=3Federate&PartnerIdpId=https://idp.abac.test:9031&Subject=lsmith@abac.test

- **Attribute Query URL**: the URL specifying the endpoint inside RP (Relying Party) that will query the IDP, i.e., https://rp.abac.test:9031/sp/startAttributeQuery.ping
- **AppId field used in query**: the unique identity of the initiating application, i.e., appid
- **Shared Secret field used in query**: used to authenticate the initiating application. The AppId and SharedSecret must both match the application authentication settings within the PingFederate server, i.e. !23234Federate
- **Partner IDP ID**: used to identify the specific IdP partner to which the Attribute Query should be sent. If this parameter is not present, the Subject and Issuer are used to determine the correct IdP, i.e., https://idp.abac.test:903
10.4 NextLabs PIP Plugin

10.4.1 Architecture

The NextLabs Control Center can support custom PIP plugin extensions for dynamic user and resource attribute retrieval during runtime. In order to install and deploy a PIP plugin such as the one described in this section, it is necessary to have previously installed and deployed the NextLabs Control Center, Policy Controller, Policy Studio, and the NextLabs Entitlement Manager (Chapter 7).

According to the NextLabs PDP Policy Extension documentation, which is only available to NextLabs customers at this time, one method for leveraging this PIP extension capability is by way of a getAttribute() function within a UserAttrProviderMod class. The PIP Plugin implements methods defined by the ISubjectAttributeProvider interface. The ISubjectAttributeProvider interface declares the method getAttribute() function which enables querying for a single subject attribute sequentially until all missing required attributes have been requested.

10.4.1.1 Required classes of the NextLabs PIP Plugin:

- UserAttrProviderMod class must exist and must contain a getAttribute() function.
  - The getAttribute() function must accept two arguments (IDSubject and String) and return an EvalValue. The EvalValue is created using its build() function and the attribute value ultimately returned from the Protocol Broker (see section 10.5, Protocol Broker).

- HTTPSTransmitter class
  - makes an HTTPS request to the Protocol Broker using a doPost() function

- CacheKey class, implementing a local Ehcache
• The CacheKey class constructor takes two parameters, the subjectId and the attributeName, which serve as a compound cache key for storing and retrieving the value of a given user's attribute within the plugin's local Ehcache.

10.4.1.2 Other Required Files or Deployment Notes:

• The three above classes must be compiled into a .jar file.

• Our method of compilation in this build was using Apache Maven 3.2.5. Maven compilations are directed by a pom.xml ("Project Object Model"), which is an XML representation of a Maven project. More information about Apache Maven and its pom file requirements can be found here: https://maven.apache.org/pom.html

• According to NextLabs support, be sure to include within the pom.xml file configuration a statement that specifies the Provider-Class. The Provider-Class is the UserAttrProviderMod class that contains the getAttribute() method. Example pom.xml excerpt from the pom.xml file in this implementation:

  <configuration>
    <archive>
      <manifest>
        <mainClass>nist.pdpplugin.UserAttrProviderMod</mainClass>
      </manifest>
      <manifestEntries>
        <Provider-Class>nist.pdpplugin.UserAttrProviderMod</Provider-Class>
      </manifestEntries>
    </archive>
  </configuration>

• Also required per NextLabs support documentation, for any custom plugin you must include a .properties file.

• The configuration file should end with the "properties" file extension. Example from this implementation: nlsampluginService.properties

• Contents should be similar to our example copied below. You must include a category = ADVANCED CONDITION statement per NextLabs deployment and loading requirements:

    name = NLSAMLPlugin_Service
    jar-path = [NextLabs]/Policy Controller/jservice/jar/nlsamplugin/NLSAMLPlugin-0.0.1-SNAPSHOT-jar-with-dependencies.jar
    friendly_name = NLSAMLPlugin Service
description = NLSAMLPlugin Service
10.4.1.3 Notes on Jar and Properties File Deployment within NextLabs Policy Controller

Software Architecture:

- The jar file containing the three classes must be deployed on the SharePoint server within the NextLabs Policy Controller software architecture in a specific location. Under the C:/Program Files/NextLabs/Policy Controller/jservice/jar folder you must create a folder specifically for your custom jar, i.e., C:/Program Files/NextLabs/Policy Controller/jservice/jar/custom_jar_folder_you_create.

- Any other required supporting jars can be compiled within the same jar as the UserAttrProviderMod class and other classes deployed as described in the previous step.
  
  - Otherwise, any additional required supporting jars can be compiled into a separate jar which is deployed elsewhere within the NextLabs Policy Controller software architecture on the SharePoint server, i.e., C:/Program Files/NextLabs/Policy Controller/jre/lib/ext/

- The properties file must be deployed on the SharePoint server within the NextLabs Policy Controller software architecture in a specific location, under the C:/Program Files/NextLabs/Policy Controller/jservice/config folder, i.e., C:/Program Files/NextLabs/Policy Controller/jservice/config/jarpropertiesfile.properties.

10.4.2 Understanding how the NextLabs PIP Plugin interacts with Build Components

When a policy is executed and the NextLabs Policy Controller PDP determines that attributes sent in the initial set up of the session are insufficient, the.getAttribute() function in the UserAttrProviderMod within the NextLabs Plugin jar is automatically executed sequentially for each missing attribute.

As described above, when the initial set of attributes is insufficient, the NextLabs PIP Plugin first checks a local cache, implemented using the Ehcache library and a CacheKey class illustrated above. If the requested attribute exists within the local cache, the NextLabs PIP Plugin retrieves and returns it immediately for use during policy evaluation by the Policy Controller (PDP).

If the requested attribute does not exist within the local cache, the NextLabs PIP Plugin’s HTTPSTransmitter class makes an https request to the Protocol Broker using a doPost() function. The Protocol Broker performs its functions and returns either the desired attribute or an exception back to the NextLabs PIP Plugin, where the Policy Controller (PDP) can evaluate the relevant ABAC policy and determine an access decision. In the case that the requested attribute does not exist, the NextLabs Policy Controller PDP is configured to default to Deny access in our build. The NextLabs Policy Controller PDP is also configured to Deny Access whenever the Protocol Broker or the NextLabs PIP Plugin produces an exception.
10.4.3 Compilation and Deployment

10.4.3.1 Compiling the NextLabs PIP Plugin Jar

1. Verify that you are on the server hosting your SharePoint instance, called the SharePoint server in our build.

2. Click on the Windows icon and begin typing **Cmd**.

3. Double-click the icon to open the Command Prompt.

4. In the Command Prompt window, navigate to the folder where your pom.xml exists and click **Enter**, i.e., `cd C:/software/java/plugin/`

5. In the Command Prompt window, run the following command and press **Enter** to compile your files and jar(s) into a single jar: `mvn clean install`

10.4.3.2 Stopping the NextLabs Policy Controller Service Before NextLabs PIP Plugin Jar Deployment

1. Still on the SharePoint server, click on the Windows icon and begin typing **Services**.

2. Double-click the icon to open the Services application.

3. In the Services application window, in the list of services, click on the **Name** column to sort by alphabetical order and look for **Control Center Enforcer Service**.
4. If the status of the Control Center Enforcer Service is running, stop it by following these steps:
   a. Click on the Windows icon.
   b. On your main screen, double-click the Stop Policy Controller shortcut.
   c. Enter your NextLabs Administrator credentials, then click Stop.
   d. Click OK.

10.4.3.3 Deploying the NextLabs PIP Plugin Jar and its Configuration File

1. Still on the SharePoint server, Click on the Windows icon and begin typing Cmd.
2. Double-click the icon to open the Command Prompt.
3. In the Command Prompt window, navigate to the folder where your NextLabs Policy Controller installation exists, and into its /jservices/jar folder where custom plugins are required to be stored, then press Enter. i.e., cd C:/Program Files/NextLabs/Policy Controller/jservice/jar/
4. In the Command Prompt window, enter a command similar to the following and press Enter to create an empty folder named after your plugin: mkdir nlsamlplugin
5. In the Command Prompt window, enter a command similar to the following and press Enter to copy your plugin jar from its existing location (example: C:/software/java/plugin/target/) to the new plugin folder you just created: copy "C:/software/java/plugin/target/plugin.jar" "nlsamlplugin/"

6. In the Command Prompt window, enter a command to navigate to the folder where your NextLabs Policy Controller installation exists, and into its jservices folder which contains the config folder where custom plugin .properties files are required to be stored, then press Enter. i.e., cd C:/Program Files/NextLabs/Policy Controller/jservice/

7. In the Command Prompt window, enter a command similar to the following and press Enter to copy your plugin .properties file from its existing location (example: C:/software/java/plugin/) to the config folder: copy "C:/software/java/plugin/nlsamlpluginService.properties" "config/"

10.4.3.4 Resetting IIS and Restarting the NextLabs Policy Controller Service

1. Click on the Windows icon and begin typing PowerShell.
2. Double-click the icon to open Windows PowerShell.
3. In the Windows PowerShell window, type in this command and press Enter to reset Internet Information Services: iisreset
4. Click on the Windows icon and begin typing Services.
5. Double-click the icon to open the Services application.
6. Within the Services application window, in the list of services, click on the Name column to sort by alphabetical order and look for Control Center Enforcer Service.
7. Right-click Control Center Enforcer Service and click Start.
   - It may be necessary to click the Refresh icon in order to see the Control Center Enforcer Service status change to running.

10.5 Protocol Broker

10.5.1 Architecture

The Protocol Broker decouples communication between the NextLabs Plugin and PingFederate RP. As noted earlier, the Protocol Broker is a web application hosted on a tomcat server installed on the SharePoint server. It communicates using mutual TLS and listens on the localhost. This ensures that the service provided by Protocol Broker is not available on the network, and the requester must be authenticated during each request.

SAMLProxy extends the HttpServlet class, which is an abstract class. This enables SAMLProxy class to read/write the http request/response, and determines the http method of the request (i.e. HTTP GET, POST, PUT, DELETE, HEAD etc) and calls one of the corresponding methods. The SAMLProxy class only implements the POST method.
The SAMLProxy class constructs an object of the SoapHTTPTransmitter class. This class reads **abacClient.jks** and **truststore.jks** which are used for mutual TLS communication initiated by the SoapHTTPTransmitter with PingFederate. It also reads **abacSigningClient.jks**, which is used to sign the SAML AttributeQuery, and metadata to verify the SAML Response signature. The jks extension stands for Java Key store, which is a storage facility for cryptographic keys and certificates.

The Protocol Broker facilitates secure communication between the NextLabs PIP Plugin and PingFederate RP. This coordination consists of two parts:

1. Communication between the NextLabs PIP Plugin and the Protocol Broker
2. Communication between the Protocol Broker and the PingFederate RP server

### 10.5.1.1 Communication Between NextLabs PIP Plugin and Protocol Broker

The Protocol Broker's **doPost()** method expects the following parameters:

- Requester
- SubjectId
- AttributeName

On successful receipt of a request, SAMLProxy uses the SoapHTTPTransmitter class to transmit the request to the PingFederate RP server. The response received from SOAPHTTPTransmitter is dispatched back to the NextLabs PIP Plugin, which then hands the result off to the PDP for policy evaluation and access decision making.

### 10.5.1.2 Communication Between Protocol Broker and PingFederate RP Server

The PingFederateRP and ProtocolBroker communicate using Assertion Query/Request Profile. As shown in **figure 10.6, Communication Between Plugin and Relying Party**, Protocol Broker initiates the secured communication on a mutual TLS channel with the Relying Party, and sends a signed SAML2 AttributeQuery. The message format and structure of the AttributeQuery is defined by SAMLCore section 3.3.2.3. Binding for the profile is defined by SAMLBind section 3.2.3. Processing rules governing the profile are provided by section 3.3 of SAMLCore. In response, Protocol Broker expects a SAML response back.

OpenSAML is used to implement an Assertion Query/Request Profile. OpenSAML is a set of open source libraries meant to support developers working with Security Assertion Markup Language (SAML). The configuration required to use the OpenSAML library is provided in section 10.5.2.2.
Figure 10.6 Communication Between Plugin and Relying Party

Based on keystores and configuration read during initialization, SoapHTTPTransmitter creates a SAML2AttributeQueryBuilder class to build a Signed SAML 2.0 Attribute Query. Attribute names received earlier in the doPost() method are used to build the AttributeQuery. A SOAPSAML2 object is used to provide SOAP parameters for the SAML message created earlier. It reads SAML 2.0 metadata to find the location of the Attribute Authority end point. It uses HttpSOAPClient to dispatch the request to the end point using mutual TLS.

HTTPSoapClient is also responsible for receiving the Attribute response, verifying the signature and sending the attributes back to the Nextlab Plugin.

Figure 10.7 Protocol Broker Interaction Diagram
10.5.2 Deployment

10.5.2.1 System and Environment Requirements

The Protocol Broker is deployed on tomcat 8.0.22 on the SharePoint server, and uses OpenSAML 2.6.4.

10.5.2.2 Configuration

In order to accept traffic only on the channel protected by mutual TLS:

1. Install tomcat on the SharePoint server. The tomcat installation procedure is provided here.

2. Open the configuration file server.xml inside the configuration directory of the tomcat installation. Comment out the section:

```xml
<!--
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
-->
```

3. Update/insert the following line:

```xml
    SSLEnabled="true" scheme="https" secure="true"
    keystoreFile="C:\Users\name\Documents\softwares\tomcat\apache-tomcat-8.0.22\conf\abacTomcat.jks" keystorePass="....password" clientAuth="true"
```
The configuration details for OpenSAML are provided here. In this demonstration, a folder called **endorsed** is created inside the **lib** directory of tomcat installation.

Add the following libraries to the endorsed folder created in the above step:

- xml-apis-2.10.0.jar
- xml-resolver-1.2.jar
- xercesImpl-2.10.0.jar
- xalan-2.7.1.jar
- serializer-2.10.0.jar

10.5.2.3 Preparation and Compilation

In our build, we used Apache Maven for Protocol Broker compilation. In order to prepare and compile the Protocol Broker, follow these steps:

10.5.2.3.1 Preparation

1. On the SharePoint server, click on the Windows icon and begin typing **Cmd**.
2. Double-click the icon to open the Command Prompt.
3. In the Command Prompt window, navigate to the folder where your pom.xml for the Protocol Broker exists, and press Enter. i.e., `cd C:/software/java/samlNewPlugin/`
4. Type the following command, then press Enter to prepare for compilation of the new Protocol Broker:
   ```
   .war file: mvn clean
   ```
5. Verify that your results are similar to the following, including the **Build Success** statement:

```
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------------------------------------------------------------------
[INFO] Building SAMLProxy 0.0.1-SNAPSHOT
[INFO]
[INFO] --------------------------------------------------------------------------------
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ SAMLProxy ---
[INFO] Deleting /home/sjha/pdpPlugins/SAMLProxy/target
[INFO]
[INFO] BUILD SUCCESS
[INFO]
```
10.5.2.3.2 Compiling the .war File

1. After following the instructions above to prepare for compiling, within the Command Prompt window, enter the following command and press Enter to create the Protocol Broker: .war file: mvn package

2. Verify that your results are similar to the following, including the Failures: 0 and Build Success portions:

   - [INFO] Scanning for projects...
   - [INFO]
   - [INFO]
   - [INFO] Building SAMLProxy 0.0.1-SNAPSHOT
   - [INFO]
   - [INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ SAMLProxy ---
   - [INFO] Using 'UTF-8' encoding to copy filtered resources.
   - [INFO] Copying 9 resources
   - [INFO]
   - [INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ SAMLProxy ---
   - [INFO] Nothing to compile - all classes are up to date
   - [INFO]
   - [INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ SAMLProxy ---
   - [INFO] Using 'UTF-8' encoding to copy filtered resources.
   - [INFO] Skip non existing resourceDirectory
   - [INFO] /home/sjha/pdpPlugins/SAMLProxy/src/test/resources
   - [INFO]
   - [INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ SAMLProxy ---
   - [INFO] Nothing to compile - all classes are up to date
   - [INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ SAMLProxy ---

[INFO] Surefire report directory:
/home/sjha/pdpPlugins/SAMLProxy/target/surefire-reports

-------------------------------------------------------
T E S T S
-------------------------------------------------------

Running nist.pdpplugin.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.03 sec

Results:

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]

[INFO] --- maven-war-plugin:2.6:war (default-war) @ SAMLProxy ---
[INFO] Packaging webapp
[INFO] Assembling webapp [SAMLProxy] in
[/home/sjha/pdpPlugins/SAMLProxy/target/SAMLProxy-0.0.1-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources
[/home/sjha/pdpPlugins/SAMLProxy/WebContent]
[INFO] Webapp assembled in [440 msecs]
[INFO] Building war:
/home/sjha/pdpPlugins/SAMLProxy/target/SAMLProxy-0.0.1-SNAPSHOT.war
[INFO]

-------------------------------------------------------------------
[INFO] BUILD SUCCESS
-------------------------------------------------------------------

[INFO] Total time: 6.281 s
[INFO] Final Memory: 11M/26M

-------------------------------------------------------------------
10.5.3 Example SAML Request and Response Output

10.5.3.1 Example of Tomcat Output from our Build that Illustrates a SAML Request

```xml
<saml2p:AttributeQuery ID="_7a41be2e3d0d1abead3e857a80b3cfbc"
  xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/">
  <saml2:Issuer
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
      <ds:SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#_7a41be2e3d0d1abead3e857a80b3cfbc">
        <ds:Transforms>
          <ds:Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>hz3JxkkIsCL/BVlkRCrgUykjbho=</ds:DigestValue>
      </ds:Reference>
      <ds:KeyInfo>
        <ds:RSAKeyValue>
          <ds:Modulus>uzxrL5iAIpNyEXHmGTDW1mzx77YJa1/c9Ruxag3sifjzuUdBjEznFJJaagM2pzTU5JCaLzgm71V
            SBmuVL+6PzTxReM3i5XzWjpRMiizadnQ0wmCryKuNaQiBFILoMbi+ySdvu+M/xhHLRxuFjY9N</ds:Modulus>
        </ds:RSAKeyValue>
        <ds:Exponent>65537</ds:Exponent>
      </ds:KeyInfo>
    </ds:SignedInfo>
  </ds:Signature>
</saml2p:AttributeQuery>
```
10.5.3.2 Example of Tomcat Output from our Build that Illustrates a SAML Response

```xml
<?xml version="1.0" encoding="UTF-8"?><S11:Envelope xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/">
  <S11:Body>
    <samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="LkF9NevJONpgbE56hszf50kpvT4qupCnXVKVqzDPZON0XcJcWwSi9pRvkGtVxqhh282ODYzc13nkpsg15F8hr7kOjQ=="><ds:Modulus>
      <ds:Exponent>AQAB</ds:Exponent>
    </ds:RSAKeyValue>
  </samlp:Response>
</S11:Body>
</S11:Envelope>
```

```xml
<saml2:Subject xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">jdoe</saml2:NameID>
</saml2:Subject>

<saml2:Attribute Name="firstname" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"/>
</saml2p:AttributeQuery>
```
10.6 Apache Directory Service (ApacheDS)

ApacheDS is included in Apache Directory Studio, which has multiple functionalities with ApacheDS Server, i.e., LDAP Browser, Schema Editor, Apache Configurator, LDIF Editor, Embedded ApacheDS, and ACI Editor.
10.6.1 Layout

Before installation, it is important to consider system needs and match them with the installation layout. The general layout for ApacheDS consists of two major concepts:

1. Installation Layout: The installation is where all files essential to ApacheDS are stored, i.e., launch script, libraries, and a service wrapper (depending on the kind of installer used).

2. Instance Layout: ApacheDS is built to run multiple instances of the server at the same time, which means that an optional instances folder can be found in the installation layout (or elsewhere on the disk, depending on the platform). In that folder you will find one or multiple directories, all sharing the same layout, corresponding to all ApacheDS instances (one directory per instance, with names corresponding to the ID of the instance).

A detailed discussion of these concepts can be found here.

10.6.2 Download

ApacheDS can be downloaded as binary or as source, and compiled on a given platform. Source can be downloaded here.

In this project, ApacheDS was downloaded as a packaged Windows installer from this location. Native installers are available in the following formats, and their download links are available at following site.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Installer Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>.exe</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>.dmg</td>
</tr>
<tr>
<td>Debian</td>
<td>.deb</td>
</tr>
<tr>
<td>Linux</td>
<td>.rpm, .bin</td>
</tr>
</tbody>
</table>

1. At the download location, you will see a URL as shown in the example below. Click the link above to download Apache Directory Server for Windows.
2. During the software download, different installation graphics will be displayed depending on which browser you use. Example from Windows Internet Explorer:

3. On Chrome, it may display as below (if you are not using command line tools):

10.6.2.1 Verify the Integrity of the Downloaded File

It is essential to verify the integrity of the file when the download completes.

The file's integrity can be verified with PGP signatures using PGP or GPG. First, download the KEYS and the asc signature file for the relevant distribution. Both KEYS and asc can be found to the right of the download link, as shown in Figure 4: ApacheDS download.

Verify the signatures using the following commands in the Command Prompt:

```bash
$ pgpk -a KEYS
$ pgpv apacheds-2.0.0-M20.exe.asc
```

or

```bash
$ pgp -ka KEYS
$ pgp apacheds-2.0.0-M20.exe.asc
```

or

```bash
$ gpg --import KEYS
$ gpg --verify apacheds-2.0.0-M20.exe.asc
```

Alternatively, you can verify the MD5 signature on the files. A Unix program called md5 or md5sum is included in many Unix distributions. It is also available as part of GNU Textutils. Windows users can get binary md5 programs from here, here, or here.

10.6.3 Installation

Note: To install ApacheDS as a Windows service, you need administrative privileges. We installed ApacheDS on Windows Server 2012. The ApacheDS installation procedure for other operating systems can be found here.

1. Once ApacheDS is downloaded and verified, double-click the installer to open it. Note: It may have already been opened by your web browser.

2. When the following screen appears, click Next.
3. Review the License agreement and click I Agree.

4. The next screen prompts you for the install path. In our build, we left the default install path. Specify an install path of your choosing, and click Next.
5. Specify a location for storing ApacheDS instances, then click **Next**.

6. The next screen asks for the location of your Java runtime environment (JRE). It is assumed, based on the earlier description in section 10.8.2, that users will have the proper Java environment prior to attempting to install ApacheDS. Users who have no JRE installed should abandon the install by clicking **Cancel**. Install the JRE and re-run the ApacheDS install. We accepted the default as shown.
7. Click **Install**. Once the installation is complete, you will receive the following prompt:
10.6.3.1 Functional Test of the ApacheDS Installation

1. Click **Show Details** in above diagram to see details of installation. Make sure all of the folders exist, then click **Next**.

2. Click **Finish** to end the installation.

3. Click **Yes** to start the ApacheDS server. Instructions are provided in section 6.2 of this chapter.
10.6.4 Starting and Stopping the Server

The server can be started and stopped with the Windows Services manager (Control Panel -> Administrative Tools -> Services). The user must have administrative privileges.

From here, ApacheDS can be started, stopped, or restarted.

The process for starting and stopping ApacheDS on other operating systems is described here.

10.6.5 ApacheDS Configuration

ApacheDS Server and Schema configuration details are provided here.

10.7 PingFederate - Apache Integration

This section requires knowledge of the following pieces of information:

1. Server IP address or hostname
2. Server port where it is listening on
3. Server credentials (i.e., private key and certificate) to be provision
10.7.1 Provisioning of Server Credential

Start Apache Directory Server Studio and open a new connection.

10.7.1.1 Creation of Server Connection

To create a new LDAPS connection, complete the following steps:

1. Define network parameters.
2. Define authentication parameters.
3. Define additional browser options (optional).
4. Define additional edit options (optional).
5. Once a new connection is opened, the following screen appears. Fill in Hostname and Port. Select the encryption method Use SSL encryption(ldaps://), then click Next.

![New LDAP Connection Screen](image)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection name</td>
<td>The name of the connection. In the Connections view, the connection is listed with this name. The name must be unique.</td>
<td>empty</td>
</tr>
<tr>
<td>Hostname</td>
<td>The hostname or IP address of the LDAP server. A history of recently used hostnames is available through the drop-down list.</td>
<td>empty</td>
</tr>
<tr>
<td>Port</td>
<td>The port of the LDAP server. The default port for non-encrypted connections is 389. The default port for ldaps:// connections is 636. A history of recently used ports is available through the drop-down list.</td>
<td>10636</td>
</tr>
<tr>
<td>Encryption method</td>
<td>The encryption to use. Possible values are: No encryption, ldaps:// and StartTLS extension.</td>
<td>No encryption</td>
</tr>
<tr>
<td>Provider</td>
<td>Option to choose either JNDI or Apache Directory LDAP client API</td>
<td></td>
</tr>
<tr>
<td>Check network parameter</td>
<td>Use this function if you want validate that the entered information is correct, and the server is reachable.</td>
<td></td>
</tr>
<tr>
<td>Read-Only</td>
<td>If this option is chosen, any attempts to modify will return an error.</td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 10. Configuring a Secondary Attribute Provider

#### Option | Description | Default
--- | --- | ---
**Authentication Method** | Select your authentication method:  
- Anonymous Authentication: connects to the directory without authentication.  
- Simple Authentication: uses simple authentication using a bind DN and password. The credentials are transmitted in clear-text over the network.  
- CRAM-MD5 (SASL): authenticates to the directory using a challenge-response authentication mechanism. The credentials are not transmitted in clear-text over the network.  
- DIGEST-MD5 (SASL): another challenge-response authentication mechanism. Additionally, you could define your realm and QoP parameters.  
- GSSAPI (Kerberos): user Kerberos-based authentication. Additional parameters can be defined. | Simple Authentication
**Bind DN or user** | The distinguished name or user ID used to bind. Previously entered DNs can be selected from drop-down list. | empty
**Bind Password** | The password used to bind. | empty
**Save password** | If checked, the password will be saved in configuration. If not checked, you must enter the password whenever you connect to the server. Warning: The password is saved as plain text. | checked
**Check Authentication** | Use this function to attempt a connection plus a bind to the host upon completion of the wizard. It will validate that the entered information is correct. | ---
This project does not use SASL or Kerberos.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get base DNs from Root DSE</td>
<td>If checked, the base DNs are fetched from the namingContexts attribute of the Root DSE.</td>
<td>checked</td>
</tr>
<tr>
<td>Fetch Base DNs</td>
<td>Use this function to get the namingContext values from the Root DSE. The returned values will appear in the Base DN drop-down list.</td>
<td>-</td>
</tr>
<tr>
<td>Base DN</td>
<td>The Base DN to use. You may enter a DN manually or select one from the drop-down list. This field is only enabled if the option Get base DNs from root DSE is off.</td>
<td>empty</td>
</tr>
<tr>
<td>Count Limit</td>
<td>Maximum number of entries returned from the server when browsing the directory. It is also used as default value when searching the directory. A value of 0 means no count limit. Note that this value is a client-side value. It is also possible to use a server-side limit.</td>
<td>1000</td>
</tr>
<tr>
<td>Time Limit</td>
<td>The maximum time in seconds the server searches for results. This is used as default value when browsing or searching the directory. A value of 0 means no limit. Note that this value is a client-side value. It is also possible to use a server-side limit.</td>
<td>0</td>
</tr>
<tr>
<td>Alias Dereferencing</td>
<td>Specifies whether aliases should be dereferenced while finding the search base entry, when performing the search, or both. To manage (create, modify, delete) alias objects you must uncheck both options.</td>
<td>Both finding and searching</td>
</tr>
</tbody>
</table>
Referrals Handling

Specifies the referral handling.

- **Follow Referrals Manually**: Received referrals and search continuations are displayed in the browser. When you open or expand a search continuation, the search is continued. Specify which connection you want to use to follow a specific referral URL. You will have full control regarding encryption and authentication options when following referrals.

- **Follow Referrals Automatically**: Follows referrals and search continuations immediately if they are received from the directory server. Specify which connection you want to use to follow a specific referral URL. You will have full control regarding encryption and authentication options when following referrals.

- **Ignore Referrals**: Any referral or search continuation received from the directory server is silently ignored. No error is logged, no dialog appears, no special entry is displayed in the DIT, and no ManageDsaIT control is sent to the server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use ManageDsaIT control while browsing</td>
<td>If enabled, the ManageDsaIT control is sent to the server in each request. This signals the directory server not to send referrals and search continuations, but return the special referral objects. Note: This is only applicable if the directory server supports the ManageDsaIT control.</td>
<td>unchecked</td>
</tr>
<tr>
<td>Fetch subentries while browsing</td>
<td>If enabled, both normal and subentries according to RFC 3672 are fetched. This causes additional search requests while browsing the directory.</td>
<td>unchecked</td>
</tr>
<tr>
<td>Paged Search</td>
<td>If enabled, the simple paged result control is used while browsing the directory. With page size you can define how many entries should be retrieved in one request. If Scroll Mode is enabled, only one page is fetched from the server at a time. While browsing, you can scroll through the pages by using <strong>next page</strong> and <strong>top page</strong>. If disabled, all entries are fetched from the server. The paged result control is only used in the background to avoid server-side limits.</td>
<td>unchecked</td>
</tr>
<tr>
<td>Fetch operational attributes while browsing</td>
<td>If enabled, both user attributes and operational attributes are retrieved while browsing. If the server supports the feature <strong>All Operational Attributes</strong>, use + to retrieve operational attributes. Otherwise, all operational attributes defined in the schema are requested.</td>
<td>unchecked</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Modify Mode</td>
<td>Specify the modify mode for attributes with an equality matching rule. Options:</td>
<td>Optimized Modify Operations</td>
</tr>
<tr>
<td></td>
<td>- Optimized Modify Operations: uses add/delete by default, uses replace if operation count is less</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Always REPLACE: always uses replace operations to perform entry modifications</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Always ADD/DELETE: always uses add and/or delete operations to perform entry modifications</td>
<td></td>
</tr>
<tr>
<td>Modify Mode (no equality matching rule)</td>
<td>Specify the modify mode for attributes with no equality matching rule. Options:</td>
<td>Optimized Modify Operations</td>
</tr>
<tr>
<td></td>
<td>- Optimized Modify Operations: uses add/delete by default, uses replace if operation count is less</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Always REPLACE: always uses replace operations to perform entry modifications</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Always ADD/DELETE: always uses add and/or delete operations to perform entry modifications</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recommended values for various LDAP servers:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- ApacheDS: Optimized Modify Operations or REPLACE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- OpenLDAP: REPLACE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- OpenDS / SunDSEE: Optimized Modify Operations or REPLACE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FedoraDS / 389DS: Optimized Modify Operations (missing equality matching rules for many standard attribute types)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Active Directory: Optimized Modify Operations (exposes no equality matching rules at all)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- eDirectory: Optimized Modify Operations (exposes no equality matching rules at all)</td>
<td></td>
</tr>
<tr>
<td>Modify Order</td>
<td>Specify the modify order when using add and delete operations.</td>
<td>Delete first</td>
</tr>
</tbody>
</table>


6. Go to Open Configuration for the newly created connection.
7. Make sure **Enable LDAPS Server** is checked, and **Port** is the same as provided during creation of the connection.

8. Go to **SSL/Start TLS Keystore**.

9. Provide the **location** of the Keystore file and the **password** for the certificate.

10. **Save** the configuration.

11. **Restart** the server.

### 10.7.1.2 Verification

OpenSSL was used to acquire the server public certificate.

```
> openssl s_client -showcerts -connect 10.33.7.8:10636 < /dev/null |
> openssl x509 -outform PEM > dir.pem
depth=0 C = US, O = ASF, OU = Directory, CN = battlefield.bb-abac-bb1.nccoe.lab
```
verify error:num=20:unable to get local issuer certificate
verify return:1
depth=0 C = US, O = ASF, OU = Directory, CN = battlefield.bb-abac-bb1.nccoe.lab
verify error:num=27:certificate not trusted
verify return:1
depth=0 C = US, O = ASF, OU = Directory, CN = battlefield.bb-abac-bb1.nccoe.lab
verify error:num=21:unable to verify the first certificate
verify return:1
DONE
[sjha@battlefield ~]$ more dir.pem
-----BEGIN CERTIFICATE-----
MIIBjDCCATYCBgFM1JE24DANBgkqhkiG9w0BAQUFADBCMQswCQYDVQQGEwJVUzEM
MAoGA1UEChMDQVNGMRIwEAYDVQQLEwlEaXJlY3RvcnkxETAPBgNVBAMTCEFwYWNo
ZURTMB4XDTE1MDQwNzE1NDgwN1oXDTE2MDQwNjE1NDgwN1owWzELMAkGA1UEBhMC
VVMxDDAKBgNVBAoTA0FTRjESMBAGA1UECxMJRGlyZWN0b3J5MSowKAYDVQQDEyFi
YXR0bGVmaWVsZC5iYi1hYmFjLWlXiMS5uY2NvZGZsYWItY2ANCgkqkhkiG9w0BAQw
-----BEGIN CERTIFICATE-----
Verify the certificate received from the directory server against the certificate that was loaded earlier.
10.7.1.3 Configuration Steps on PingFederate RP Server

1. The following screen will appear, displaying all certificates on the server's global trust list.

2. Select Import Certificate.
3. Choose a file to import.

4. Once your chosen file appears in the Filename field, click Next.

5. View the Summary of the imported certificate.
6. **Click Done.** The main screen will display a list of certificates. Click **Save**.

10.7.1.4 **Creation of Data Store to Connect to ApacheDS**

1. **Click on Data Stores.**
2. In the Manage Data Stores window, click **Add New Data Store**.

3. Choose **LDAP** and click **Next**.

4. Provide a **Hostname** and **LdapType**.

5. It may be necessary to configure connection pooling. It is important to select **Verify LDAPS Hostname** if the directory server certificate is bound to a hostname, and this hostname can be verified.
6. If there is any binary data, enter it in the **Binary Attribute Name Field**, and click **Add**.

7. A summary of the LDAP configuration will appear.

8. A **Summary** of the connection will appear as following. Click **Save**. You will then return to the Main Admin console.

10.8 Configuration of PingFederate to Query the JIT Cache when Responding to Secondary Attribute Requests

10.8.1 Introduction

This section will cover all the configuration steps required to enable PingFederate RP to communicate with the Secondary attribute Provider and respond to its queries. The SP connection section will cover communication channel protection and message protection. To fulfill the query request from the NextLabs PIP Plugin and Protocol Broker, PingFederate queries its local LDAP server called Just in Time (JIT) cache. Note that PingFederate RP may not have data to fulfill the query. In that case, PingFederate RP extends the query to PingFederate IdP using a unique method (Ping Data source).
A Data Store is any type of source for digitized data, i.e., database, file, stream, etc. PingFederate administration console uses this term for system settings. In the Java software platform, data source is a factory for connections to the physical data source that this data source object represents. Thus, data source is the logical manifestation of a physical data store in a java application. Due to this, the terms will be used interchangeably below.

This section provides the configuration needed to query JIT cache, i.e., creation of the data source for the LDAP Server. We have already discussed the configuration of Ping Data Source in Custom Data Store section. SP connection describes how both of these data stores are chained together to fetch the result of the attribute query.

### 10.8.2 Prerequisites

Before starting this configuration, the following steps must have already been completed:

1. How-To Guides 1-6
   a. Complete Installation of PingFederate, both RP and Idp
2. Installation and configuration of ApacheDS
3. Installation of Ping Custom Data Store
4. Availability of Ping web administration console (automatically included in the PingFederate installation from previous chapters)

### 10.8.2.1 SP Connection

As described above, PingFederate (RP) acts as an IdP for the Secondary attribute provider. In order to enable support for exchange of federation-protocol messages and provide channel protection, it is essential to configure the SP (Service Provider) connection. Note: Ping Identity’s documentation uses the term Service Provider and SP where the rest of our ABAC documentation uses the term Relying Party and RP. In this document, please consider these terms interchangeable.

The following goals are achieved by configuration of the SP connection:

a. Specification of connection and associated security protocol (i.e., TLS/SSL)
b. Specification of SAML profile t including detailed security specifications (the use of digital signatures, signature verification, XML encryption)
c. Specification of Attributes that may be sent using the SAML2 Attribute Query profile
d. Specification of Data Store(s), if agreement between Idp and SP includes sending a SAML response containing attribute values from a local data store.
10.8.2.1.1 Specification of Profile

Instructions on how to create a new connection can be found here.

1. Click on Manage on All SP in the first column on the left hand side.

2. The following screen will appear. Click on Create Connection.

3. Check the box for Browser SSO Profiles and select SAML 2.0 as protocol from the drop-down menu.

4. Uncheck Browser SSO, check Attribute Query, and click Next.
5. Choose a metadata file and click **Next**.

6. SAML2 metadata has its own **specification**. As per this specification, `KeyDescriptor` is an optional sequence of elements that provides information about the cryptographic keys that the entity uses when acting in this role. However, for message authentication and integrity, it is essential to provide the certificate so that signed messages coming from the secondary attribute provider can be verified. A relevant part of metadata is shown here:

```xml
<md:KeyDescriptor use="signing">
  <ds:KeyInfo>
    <ds:X509Data>
      <ds:X509Certificate>
        MIIE4jCCAgAwIBAgICEAMwDQYJKoZIhvcNAQELBQAwYjELMAkGA1UEBhMCVCVMxE
        ETAPBgNVBAqMGE1hcnsY5Wk5MiwiEAYDVQQHDA1bNzY2MDAwMDEyNTU5OTA1
        BU5DQ29FMQ0wCwYDVQQQLDARBQkFDMQ0wCwYDVQQQLDRBQkFDMwEwDQYJKoZI
        MTA1N0XkIAIHEAoCAQxP2NrdmlsbGUxJjAMBgNVBAoMEwFzc2VEeUS5JczAUM
        TU0xOTU0TiUEmIxIzAhBgkqhkiG9w0BCQEFQWHNqAGFAT0xOTU0TiUEmIxJ3Jn
        MIIBIjANBgkqhkiG9w0BAQEEBADQgCQxP2NrdmlsbGc2IjAMBgNVBAoMME1h
        cnNhbGlicmF0aW9uMTU5MTIzNDA3MDAwMB4XDTE2MDA4MTE1MTI1NjoyMD
        P0u363e32hIyzsf2gKizv0qY2ecD1BCNITTSa2H2STf50kpvTF9upCnXVKVq
        j9NPSEmLHY4YoLrZKw2SM/Im/3bhj/xF7q7FGHMcJH4zZrr2QpMBeencyOJ
        Vro/MfyLg1VER0pu363e32hIyzsf2gKizv0qY2ecD1BCNITTSa2H2STf50kpvT9
        j9NPSEmLHY4YoLrZKw2SM/Im/3bhj/xF7q7FGHMcJH4zZrr2QpMBeencyOJ
        Vro/MfyLg1VER0pu363e32hIyzsf2gKizv0qY2ecD1BCNITTSa2H2STf50kpvT9
        j9NPSEmLHY4YoLrZKw2SM/Im/3bhj/xF7q7FGHMcJH4zZrr2QpMBeencyOJ
        Vro/MfyLg1VER0pu363e32hIyzsf2gKizv0qY2ecD1BCNITTSa2H2STf50kpvT9
        j9NPSEmLHY4YoLrZKw2SM/Im/3bhj/xF7q7FGHMcJH4zZrr2QpMBeencyOJ
        Vro/MfyLg1VER0pu363e32hIyzsf2gKizv0qY2ecD1BCNITTSa2H2STf50kpvT9
        j9NPSEmLHY4YoLrZKw2SM/Im/3bhj/xF7q7FGHMcJH4zZrr2QpMBeencyOJ
        Vro/MfyLg1VER0pu363e32hIyzsf2gKizv0qY2ecD1BCNITTSa2H2STf50kpvT9
        j9NPSEmLHY4YoLrZKw2SM/Im/3bhj/xF7q7FGHMcJH4zZrr2QpMBeencyOJ
        Vro/MfyLg1VER0pu363e32hIyzsf2gKizv0qY2ecD1BCNITTSa2H2STf50kpvT9
        j9NPSEmLHY4YoLrZKw2SM/Im/3bhj/xF7q7FGHMcJH4zZrr2QpMBeencyOJ
        Vro/MfyLg1VER0pu363e32hIyzsf2gKizv0qY2ecD1BCNITTSa2H2STf50kpvT9
```

DRAFT
7. Verify the metadata content.
8. Click on **Configure Attribute Query Profile**.

9. Specify the list of attributes that may be returned to the SP in response to an attribute request.
10.8.2.1.2 Specify a series of data stores.

1. In the **Attribute Source Id** field, specify JIT (LDAP).

   ![Attribute Source Id field](image)

   - **Attribute Source Id**: JIT (LDAP)
   - **Attribute Source Description**: Just in Time cache source
   - **Active Data Store**: 10.33.7.8.10389
   - **Data Store Type**: LDAP

2. Specify **Attributes** for the JIT Cache.

   ![Attribute Query](image)

   - **Base DN**: ou-users,ou-system
   - **Search Scope**: Subtree
   - **Attributes to return from search**:
     - **ROOT OBJECT CLASS**: Attribute
     - **ATTRIBUTE**: Action
     - **Subject DN**
     - **employeeType**
     - **<Show All Attributes>**
     - **givenName**

   ![View Retrievable Attributes](image)
3. **Specify LDAP Filter.**

![LDAP Filter image]

4. **Verify that your data is correct.**

![Attribute Sources & User Lookup image]
5. Specify a custom **Data Store**.

![Data Store Configuration](image1.png)

6. Define a filter for extracting data from this data store.

![Filter Configuration](image2.png)

7. Based on the data elements available from this data store, select the ones pertinent to this connection. Note that these are the attributes you previously selected to return from Ping Custom Data.

![Attribute Selection](image3.png)
8. Click **Retrieve**.

9. Click on **Attribute Mapping Fulfillment**.

10. **Issuance Criteria**: PingFederate can evaluate various criteria to determine whether to issue an attribute query response. Use this optional screen to configure the criteria for use with this conditional authorization.

11. Click on **Security Policy**.
12. Check the Summary.

13. Provide Credentials for the back channel attribute request.


10.8.2.1.3 Back Channel Authentication Configuration

1. Use the default Transport Layer Authentication with SSL Client Certificate.
2. It is encouraged to use the **Anchored** verification method.

3. You will be prompted to select an **SSL Verification Certificate**. In our build, a certificate has not been previously imported. **Click on Manage Certificate.**

4. **Click Import.**

5. **Click Choose File.**

6. **Select your certificate file from the Explorer window.**

7. The file name will appear in the **Filename** field.

8. **Click Next.** This will display details of parts of certificate.
9. Check **Make this the active certificate** and click **Done**.

10. Verify the certificate.

11. Under **Action**, select **Activate**.

12. View a **Summary** of the verification.
13. Return to the **Back Channel Authentication** tab.

14. Select **Digital Signature Settings** for outgoing messages, then click **Next**.

15. Go to **Digital Signature Settings**. Click **Configure**.

16. Select **Digital Signature Settings** on incoming messages.

17. Click on **Manage Signature Verification Settings**.

18. Select the certificate(s) to use when verifying these digital signatures. When multiple certificates are chosen, each certificate is tried from the top of the list down until the signature is verified. It is assumed that signed certificates have already been imported.
not, click on Manage Certificate and complete the steps detailed earlier for importing a certificate.

19. Verify the Summary.

20. This completes the signature verification credential settings.

21. Verify the Summary.
22. **Activate** the connection and **Save**.

23. **Save** again.
10.8.2.2 IDP Connection

As an SP, you are making a connection to a partner IdP. Follow these steps to select the type of connection needed for this IdP:

1. On the right hand side of the administrative console, click **Manage All IdP under IdP Connections**.

2. Open the connection that was created in chapter 6. Click on **Connection Option**. It my default to **Browser SSO**. Additionally, select **Attribute Query** and **JIT Provisioning**.
3. Click **Next**. Verify that the information in the **General Info** tab is correct.

4. Click **Next**.

5. Click on **Configure Attribute Query Profile**.
6. Specify an Attribute Authority Service URL.

7. Attributes requested by your application may not match exactly the attributes supplied by the IdP. Specify the mapping between these sets of attributes.

8. Select Sign the Attribute Query.

9. Verify that the Summary is correct, then click Done.
10. When the following screen appears, click **Next**.

11. JIT provisioning details have been provided by PingFederate [here](#).

12. **Save** the configuration.

13. Select **Application Authentication**.
14. Enter **appid** in the ID field, and use the shared secret that you input during custom data store configuration, then save the configuration.

15. Select **Browser SSO** and **Attribute Query**.

### 10.9 ApacheDS Schema Extension

At a high level, LDAP Schema is the collection of attribute type definitions, object class definitions, and other information which a server uses to determine how to match a filter or attribute value assertion (in a compare operation) against the attributes of an entry, and whether to permit add and modify operations. For a more formal definition, look into section 4.1 of RFC 4512.

ApacheDS comes with a comprehensive set of predefined, standardized schema elements. Specification of many of these elements can be found in RFC 4519. Generally, these predefined schema satisfy most of the needs of a project. However, you may sometimes be required to define additional attributes or object classes that are not included in the server provided schema.

Each attribute and object class has an associated unique Object Identifier. Generally, An Object Identifier is a tree of nodes where each node is simply a sequence of digits. The rules roughly state that once an entity is assigned a node in the Object Identifier (OID) tree, it has sole discretion to further delegate sub-trees off of that node. Some examples of OIDs include: 1.3.6.1 - the Internet OID, 1.3.6.1.4.1 - IANA-assigned company OIDs. It is formally defined using the ITU-T's ASN.1 standard, X.690.

The IANA OID registry contains a list of registered entities that use OIDs to reference internal structures. In this chapter, we have used OIDs that are not registered anywhere. For this reason, we are using the subtree 2.25, as per recommendation by ITU. UUID is generated by the program found [here](#).

In the following section, we will demonstrate how to create an attribute. Similar procedures can be used to create many attributes and object classes.

#### 10.9.1 Pre-Requisites

For Schema extension, this project used ApacheDS studio. ApacheDS installation and configuration is detailed in section 10.6 of this guide.

#### 10.9.2 Procedure

1. Start ApacheDS Studio from the Start menu.
2. The following screen will appear:

![Image of a software interface with dropdown menus and options]

4. Select the **New Schema Project** wizard.

5. Specify a **Project name**, i.e., `nist.nccoe.abac` in our build.
6. Select **Offline Schema**, then click **Next**. On the next screen, **Choose the 'core' schemas to include**.

7. Click **File -> New** and select **New Schema**.
8. Specify a Schema name, i.e., nist.nccoe.abac in our build.

9. The following screen will appear:
10. Select **Attribute Types -> New -> New Attribute Type.**

11. In the new window, choose the **OID** from the previous instructions.
12. Click **Next** to choose the superior type of this attribute.

13. Specify **Matching Rules**. Since it is a string, case insensitivity is chosen in our build.
14. The following screen will appear:
15. You can create other attributes by following process described above.

16. Export the schema by selecting Export -> Schemas for ApacheDS. It will create an LDIF file.
17. LDIF files are specified by their own RFC. In a text editor, it displays as following:
18. To import the file, first select **Window -> Open Perspective -> LDAP**.
19. Click on the left bottom corner of the window and select **New Connection**.
20. Fill in the network parameters and click **Next**.

![Network Parameter](image)

21. Provide credentials and click **Finish**.

![Authentication](image)
22. Open Schema Editor Browser and import the LDIF file created in the previous step.

23. Click Finish.
24. To verify success, the log file generated at the end of the import should show RESULT OK.

10.10 Functional Tests

Once all requirements have been met and all steps in this How-To Guide have been executed, a few functional tests will ensure that the key components of this How-To Guide were correctly deployed and are communicating with other ABAC components as desired.

The first functional test will check the ready state of the NextLabs Policy Controller (ensures that it is running after being paused for plugin deployment).

The second test will check that the plugin was successfully loaded into the NextLabs software architecture, that an attribute request is sent to the Protocol Broker from the NextLabs PIP plugin’s `getAttribute()` function, and that the Protocol Broker responds with an expected attribute value.

The second functional test will ensure that the Protocol Broker is successfully loaded and deployed within the tomcat server instance.

Both of these functional tests can be done on the SharePoint server.
10.10.1 Testing the Ready State of the NextLabs Policy Controller Service

1. Click on the Windows icon and begin typing the word Services.
2. When the Services application icon appears, double-click to open the Services application.
3. Within the Services application window, click on the Name column and look for Control Center Enforcer Service.
4. Verify that the status column reads Running.

10.10.2 Test the Successful Loading of the Custom Plugin within the NextLabs Policy Controller Software Architecture

1. Click on the Windows icon.
3. Click on the Windows Explorer application icon.
4. Navigate to C:/Program Files/NextLabs/Policy Controller/agentLog/.
5. Within the agentLog folder, note the Agentlog0.0 file.
6. Within the agentLog folder, copy and paste the locked file Agentlog0.log0 to open it for review.
   - Left-click on the file name, and hold down Ctrl+C.
   - Left-click anywhere in the agentLog folder, right-click and hold down Ctrl+V.
7. Double-click the Agent0.log-Copy.0 file to open it in your default text editor.
8. Within your default text editor, use a search function to search for standard NextLabs logging terminology to verify that the plugin was loaded correctly. Example:
   Jul 13, 2015 4:59:21 PM
   com.bluejungle.pf.domain.destiny.serviceprovider.c A
Chapter 10. Configuring a Secondary Attribute Provider

FINE: Loading C:\Program Files\NextLabs\Policy Controller\./service/config/nlsamlpluginService.properties

Jul 13, 2015 4:59:21 PM
com.bluejungle.pf.domain.destiny.serviceprovider.c A
FINE: Loading C:\Program Files\NextLabs/Policy Controller/service/jar/nlsamlplugin/NLSAMLPlugin-0.0.1-SNAPSHOT-jar-with-dependencies.jar

Jul 13, 2015 4:59:22 PM
com.bluejungle.pf.domain.destiny.serviceprovider.ServiceProviderManager register
INFO: A new Service 'NLSAMLPlugin_Service' is registered.

9. Within your default text editor, use a search function to search for logging statements you included in your plugin code to verify that the init() methods are called while the jar is loaded within NextLabs (standard according to NextLabs support). Example:

Jul 13, 2015 4:59:21 PM
gov.nist.NLSAMLPlugin.UserAttrProviderMod init
INFO: NLSAMLPlugin UserAttrProviderMod code -- init method
Jul 13, 2015 4:59:21 PM
gov.nist.NLSAMLPlugin.HTTPSTransmitter init

• You can copy and paste the locked file, or keep a live annotating tool open that will display the contents of Agent0.log0 as new log statements are recorded. Example from this implementation: BareTail by Bare Metal Software Pty Ltd.

• Example screenshot using BareTail to open the Agent0.log0 file, with optional highlighting illustrating evaluated policies in yellow:

![Example screenshot using BareTail to open the Agent0.log0 file](image-url)
10.10.3 Testing that the Protocol Broker .war File Loads Correctly in Tomcat Server

1. On the SharePoint Server, open Services, and ensure that the Control Center Enforcer Service is listed as Running.

2. Using Windows Explorer, navigate to your Apache tomcat installation within the Windows file structure. Example: `C:/software/apache-tomcat-7.0.61`

3. Double-click to open the bin folder. Example: `C:/software/apache-tomcat-7.0.61/bin`

4. Double-click `startup.bat` to start the bat, and wait for startup to complete.

5. From any computer connected to this network, open an Internet browser.

6. In the address field, type `https://sharepoint.abac.test/` and press Enter.
7. Choose **Federated Logon** from the drop-down menu.

8. At the login screen, enter the credentials of a user that exists in your IdP Active Directory (Chapter 2), and click **Sign On**.
9. Verify that the user was able to access the main page of the RP's SharePoint. Example:

10. In the SharePoint site, double-click on an object for which you know the user will be missing an attribute in order to be granted access, but that can be retrieved via a secondary attribute request using the NextLabs PIP plugin, Protocol broker, and Ping custom data store.

11. Follow the remaining steps 15-18 to verify through standard and custom logging that the Protocol Broker was loaded, that the getAttribute() from the NextLabs PIP plugin was sent, and an expected attribute value was returned.

12. In Windows Explorer, navigate to your installation of Apache tomcat and locate its log files, i.e., C:/software/apache-tomcat-7.0.61/logs

13. Open a catalina.___.log file using your default text editor and use a search function to find standard Apache tomcat logging that indicates the .war file was correctly deployed and loads without error. For example, in C:/software/apache-tomcat-7.0.61/logs/catalina.2015-06-29.log:

Jun 29, 2015 1:49:16 PM
org.apache.catalina.startup.VersionLoggerListener log
INFO: Server version:    Apache Tomcat/7.0.61

Jun 29, 2015 1:49:16 PM
org.apache.catalina.startup.VersionLoggerListener log
INFO: CATALINA_BASE:
C:\software\java\samlNewPlugin\apache-tomcat-7.0.61

Jun 29, 2015 1:49:16 PM
org.apache.catalina.startup.VersionLoggerListener log
INFO: CATALINA_HOME:
C:\software\java\samlNewPlugin\apache-tomcat-7.0.61
INFO: Command line argument:
-Djava.util.logging.config.file=C:\software\java\samlNewPlugin\apache-tomcat-7.0.61\conf\logging.properties
INFO: Command line argument:
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
INFO: Command line argument:
-Djava.endorsed.dirs=C:\software\java\samlNewPlugin\apache-tomcat-7.0.61\endorsed

INFO: Deploying web application archive
C:\software\java\samlNewPlugin\apache-tomcat-7.0.61\webapps\SAMLProxy-0.0.1-SNAPSHOT.war
INFO: Deployment of web application archive
C:\software\java\samlNewPlugin\apache-tomcat-7.0.61\webapps\SAMLProxy-0.0.1-SNAPSHOT.war has finished in 4,953 ms

INFO: Deploying web application directory
C:\software\java\samlNewPlugin\apache-tomcat-7.0.61\webapps\docs
INFO: Deployment of web application directory
C:\software\java\samlNewPlugin\apache-tomcat-7.0.61\webapps\docs has finished in 78 ms

INFO: Deploying web application directory
C:\software\java\samlNewPlugin\apache-tomcat-7.0.61\webapps\examples
14. While the same file is open, use another search function to find custom logging that indicates that the Protocol Broker was used for a SAML Attribute query request and response. Example custom log files from this build:

```
Jun 29, 2015 1:59:00 PM nist.pdpplugin.transport.SoapHTTPTransmitter transmit
INFO: START SoapHTTPTransmitter method. Start time: 1435600740151
Jun 29, 2015 1:59:08 PM nist.pdpplugin.transport.SoapHTTPTransmitter transmit
INFO: START SoapHTTPTransmitter method. Start time: 1435600748229
Jun 29, 2015 1:59:11 PM nist.pdpplugin.transport.SoapHTTPTransmitter transmit
INFO: END SoapHTTPTransmitter transmit Method: 1435600751682
Jun 29, 2015 1:59:11 PM nist.pdpplugin.transport.SoapHTTPTransmitter transmit
INFO: END SoapHTTPTransmitter transmit Method. Total Execution time: 11531
```

15. Within the Agent0.log0, another search function to find custom logging statements that verify from within the NextLabs Policy Controller software execution side that the plugin's getAttribute() function was called and that the requested attribute was returned.

```
a. Example from this build:
   i. user: chen@abac.test
   ii. requested attribute: clearance
   iii. expected returned value: Secret
   iv. actual returned value: Secret
```

```
getAttribute
INFO: NLSAMLPlugin UserAttrProviderMod getAttribute() function called.
```
getAttribute
INFO: START getAttribute method. Start time: 1433345957517
getAttribute
INFO: NLSAMLPlugin UserAttrProviderMod getAttribute Line00-72 -
subjectID param: schen@abac.test
getAttribute
INFO: NLSAMLPlugin UserAttrProviderMod getAttribute Line00-73 -
attributeName param: clearance
getAttribute
INFO: NLSAMLPlugin Trying to check if there exist a prior entry in
cache. -- UserAttrProviderMod Line00-79
getAttribute
INFO: NLSAMLPlugin Using soapHTTPTransmitter object and calling its
transmit() function.
getAttribute
INFO: NLSAMLPlugin UserAttrProviderMod getAttribute() Line00-114 --
attributeValue returned: Secret