Technical Lesson 3.1.2

The "Attribute Value Spacing" Pitfall

Goals:

1. Understand what the Attribute Value Spacing Pitfall, and why it is problematic[1].

Summary:
In this Lesson, you will compare two policies that have a subtle difference in the value of an **<AttributeValue>** element. You will evaluate both policies against the same request and analyze the different results.

Steps:

3.1.2.1 Inspect Permit-Policy.xml and Request-1.xml

Confirm that this Permit-Policy and Request-1 are the same as the Permit-Policy and Request-1 from Lesson 3.1.1.

3.1.2.2 Evaluate Permit-Policy against Request-1

Recall from Lesson 3.1.1 that the **<Decision>** should be "Permit". Confirm that this is the case.

3.1.2.3 Compare Permit-Policy with Permit-Policy-2

See if you notice the subtle difference. The closing tag of the **<AttributeValue>** element in Permit-Policy-2 on Line 16 does not come immediately after the value "Top Secret". The **MatchId** of the **<SubjectMatch>** (Line 15) is "string-equal"; during evaluation, this function will take into account the extra spaces after the value "Top Secret".

3.1.2.4 Evaluate Permit-Policy-2 against Request-1

Execute SimplePDP with Permit-Policy-2.xml and Request-1.xml, and output the results to Request-1_Permit-Policy-2_Result.xml. Inspect Request-1_Permit-Policy-2_Response.xml. Confirm that the **<Decision>** for "Resource-1" is "NotApplicable". It is "NotApplicable" because the value "Top Secret" in the **<AttributeValue>** in Request-1 on Line 7 is not the same as "Top Secret" with extra spaces as stated in Permit-Policy-2.

---

[1] You should be very diligent when authoring policies to avoid this problem. Also, it may be possible to construct an XSLT stylesheet to ensure that this condition never occurs.

A Note about Notation

XML elements, for XACML and data files, are written as they appear in XML documents, and are indicated in boldface text. For example: **<Policy>**.

XML attributes, for XACML and data files, are written as they appear in XML documents, and are indicated in boldface text. For example: **PolicyId**.

Values of XACML and data elements appear in double quotes. For example: “Permit”.

We introduce some terms to serve as labels for certain groups of policy elements; these terms are used to enable discussions about groups of elements as a whole. These terms appear in italics. For example: *class*.

We use labels to refer to files, directories, and data items that exist in the accompanying virtual machine. These labels are used in the style of Linux environment variables – they begin with a dollar sign ($) which is followed by the label in all caps. For example: the label $POLICY_GUIDE refers to the following path on the virtual machine, “/home/guide/policy-guide”.