Flexible CA LoA support in Argus

Mischa Sallé

1 Introduction

We need a flexible mechanism for expressing CA Levels of Assurance (LoA) in Argus policies. At the moment the extension still only entails allowing CAs classified as DOGWOOD, the Identifier-Only Trust Assurance (IOTA) profile, as an additional permitted authentication source in combination with selected (VO) attribute authorities. But in the near future, there will be more new LoAs that will need to be supported. Within the EGI and related ecosystem, this will include the assurance profiles coming from the REFEDS Assurance WG (specifically the proposed combined assurance clusters Capuccino and Espresso) as well as the EGI CheckIn assurance profiles that are scoped to the service providers within EGI.

2 Problem description

A problem with the introduction of differentiated LoA is that in the EGI and WLCG infrastructure, until now, all CAs were either allowed or not. Hence a VO is automatically allowed to use any CA LoA, also the ones that are not yet in existence and ones that are of a lower LoA than the currently existing ones.
A second problem is the lack of a generic EGI-wide mechanism for implementing policies: policies are written by the site (admin), and the way this is done is site-dependent, it can be based on a simple input file, on puppet scripts etc. In the past YAIM would result in a fully configured authorization profile based on a simple list of allowed VOs, but for Argus, no such standardized configuration of the policy is available.

3 Requirements for a flexible and future-proof solution

A number of requirements for a flexible and future-proof solution can be recognized:

1. Must handle a VO becoming eligible for CA LoA X
2. Must handle a VO becoming ineligible for CA LoA X
3. Must have a minimal performance impact
4. Must allow matching more complicated combined LoAs: e.g. allow DOGWOOD only when used in combination with 2FA, or to allow DOGWOOD only for Role=production, not for Role=VO-admin.
5. A VO must not automatically become eligible for all LoAs when new assurance profiles become enabled on the infrastructure.

4 Suggested solution

We suggest to require to combine each VO- or FQAN attribute with a single new attribute, *ca-policy-names*. This attribute will force the right type(s) of CA-LoA allowed for use by that

VO. Requiring the combination of two attributes is essentially an extension of the XACML profile being used by Argus.
The requirements in section 3 are satisfied as follows:

1. Changing the accepted set of LoA for a VO is an update of the value of the *ca-policy-names* attribute for each occurrence of that VO (or derived FQAN).
2. idem
3. Performance-wise this is a fast solution, since the extra attribute key/value pair is only parsed in case the VO or FQAN attribute matches. Setting the value of the new attribute, by determining the LoA of the CA, is something that needs to be done in any case.
4. Combining different LoAs is easily expressed, either by adding multiple *ca-policy-names* attributes, or by creating new meta-policy info files which combine the required LoA. Even for very complicated and as yet unknown combinations of LoAs, the policy does not become more complicated. Expressing CA-LoA for different other attributes is easy, since LoA is expressed in a separate attribute.
5. We propose to prevent automatically allowing VOs for all LoAs by using *explicit* verification of the policy, as further discussed in section 4.1. This is necessary in particular for requirement 2.

4.1 External checks

In order to enforce that a VO does not become automatically eligible for a new LoA, a few options are available. For removing the eligibility of a VO for a certain LoA, an explicit update of the policy will always be required.
A number of tools, checks and ways to enforce updates of the policies can be considered:

I. EGI will not include IOTA CAs via the EGI-core RPM, but (at least at first) via an extra RPM, thereby requiring a manual step for activation.
II. An automatic update script for the PAP policy can been provided, that for the transition IOTA vs non-IOTA can automatically adjust the attribute. This does not enforce but makes it easier for the system administrator.
III. An explicit check of the policy against the XACML profile, to enforce that each VO- or FQAN attribute is indeed accompanied by a ca-policy-names attribute, can be implemented for example in one the following ways:
    A. The PDP, via its PIP mechanism, could get a new PIP that verifies the policy.
    B. A consistency check could be enforced in the PAP itself.
    C. An init-script could enforce conformance of the policy.
    D. By enforcing the right Argus version by a dependency in the EGI policy packages.

    A reason for doing this test in the PDP is to have it run only when the subject in the XACML request is actually using a CA from a new profile: that would allow an update of the Argus software, without requiring an update of the policy at the same time.
IV. By sending test-jobs using an ineligible combination, monitoring can spot misconfigurations.

5 Implementation

5.1 Description of the changes to existing code

This depends on which elements of 4.1 are to be implemented. All other code is ready as a pull-request.

5.2 Estimated effort required for the changes

This depends on which elements of 4.1 are to be implemented. Only the options in 4.1 III require coding effort. If those are left out, the solution can be rolled out in a very short time.

6 Operation

6.1 Upgrade procedure from 1.7.0

This depends on the scenario(s) chosen in 4.1. In general, in order to enable IOTA support, the EGI RPM needs to be explicitly installed, and the policy in the PAP needs to be updated, preferably using the script described in 4.1 II.

6.2 Procedure to enable/disable support for a LoA for a VO

Enabling/disabling support for IOTA would consist of running the script as described in 4.1 II.

6.3 Example policy

Assuming the list of VOs eligible for IOTA / DOGWOOD consists of atlas, cms, lhcb and alice we have the following example:
Before:

```
# First resource
resource "ce"   {
    action "execute" {
        # IOTA VO FQAN
        rule permit { fqan="/atlas/pilot" }
        # non-IOTA VO name
        rule permit { vo="dteam" }
        # non-voms match: untouched
        rule permit { subject-dn="/CN=pietje puk" }
        # deny rule: untouched
        rule deny {
            subject-dn="/CN=evilhacker"
            vo="test"
        }
    }
}
```

```
# Second resource
resource "wn"   {
    action "execute" {
        # IOTA VO primary FQAN
        rule permit { pfqan="/pvier" }
        # IOTA VO alternative name
        rule permit { emi-vo="atlas" }
        # deny rule: untouched
        rule deny { vo="test" }
        # combined permit rule non-IOTA VO name
        rule permit {
            vo="test"
            subject-dn="/CN=John Doe"
        }
    }
}
```

After:

```
resource "ce"   {
    action "execute" {
        # IOTA VO FQAN
        rule permit {
            fqan="/atlas/pilot"
            ca-policy-names="policy-aspen-birch-cedar-dogwood"
        }
        # non-IOTA VO name
        rule permit {
            vo="dteam"
            ca-policy-names="policy-aspen-birch-cedar"
        }
        # non-voms match: untouched
        rule permit {
            subject-dn="/CN=pietje puk"
        }
        # deny rule: untouched
        rule deny {
            subject-dn="/CN=evilhacker"
            vo="test"
        }
    }
}

# Second resource
resource "wn"   {
    action "execute" {
        # IOTA VO primary FQAN
        rule permit {
            pfqan="/pvier"
            ca-policy-names="policy-aspen-birch-cedar"
        }
        # IOTA VO alternative name
        rule permit {
            emi-vo="atlas"
            ca-policy-names="policy-aspen-birch-cedar-dogwood"
        }
        # deny rule: untouched
        rule deny { vo="test" }
        # combined permit rule non-IOTA VO name
        rule permit {
            vo="test"
            subject-dn="/CN=John Doe"
```

```
                    ca-policy-names="policy-aspen-birch-cedar"
                }
        }
}
```