

Policies in a Generic AAA Environment

<draft-taal-aaaarch-generic-pol-01.txt>

Arie Taal, Guus Sliepen, Armijn Hemel, David Spence.

Goal of this draft

The goal of the draft will be to eventually list all the problems and things one encounters when implementing an AAA server.

- How do AAA servers interact?
 - What kind of relationships do AAA servers have?
- What are policies?
 - How do policies appear within one AAA server?
 - What are the requirements of policies?
 - If we push/pull policies, how do we represent them?

Changes since -00 draft

- Specified a BNF grammar for policies.
- Lists some problems we will encounter.
 - Conflicting policies.
 - Execution of actions: delayed or not?
 - Parallel evaluation of policies: non-deterministic results?

Policy Requirements

List of requirements for policies that seem very obvious at first sight:

- Must be able to handle arbitrary boolean expressions.
- Possibly also handle arbitrary arithmetic/string expressions.
- Must be able to interact with Application Specific Modules.
- Must be able to send other requests, and push/pull other policies.
- Must be able to attach actions to (sub)expressions.

Problems with policy requirements [1]

List of requirements for policies that are not obvious at second sight:

- Arithmetic/string expression: how far do you go? `cos()`?
`sqrt()`? regexps?
- Interaction with Application Specific Modules: For example, use SQL queries. How to generate such a query from the parameters we know? This involves needing instructions for query manipulation.
- Sending out requests is easy, but pushing/pulling is not:
 - We need a policy language.
 - We need strong security.

Problems with policy requirements [2]

- Actions are also non-trivial on closer inspection:
 - Do we execute them immediately or do we wait until the policy is completely evaluated?
 - What about actions attached to remote policies?
 - * Let remote server execute them? That might give conflicts.
 - * Execute them on local server? Actions may be depending on non-standard ASMs.

Policy Representation

- Java:
 - “Normal” Java implementations too heavy, security is a problem.
 - But smartcards have a very lightweight and secure implementation.
- XML:
 - Is really a buzzword, XML by itself does nothing at all.
 - Might be used to facilitate a grammar though.
- Another language:
 - Take an existing one.
 - Make a new one.

Making a Policy Language

We are now looking into specifying our own language (using a parser generator like for example bison), just to see what is really needed and what is to be avoided.