

Request原语是对访问请求的抽象,其定义了访问请求的语义.一个访问请求通常由经典的三元组构成:进行访问的主体(subject)、被访问的客体(object)、访问动作(action).PML中描述上述三元组的Request原语可表示为:  
r = sub, obj, act  
PML提供了自定制Request原语的能力,例如如果不需要控制到具体的资源,可以采用sub, act两元组的格式.如果主体需要采用所属域等额外信息来标识的话,可以采用domain, sub, obj, act四元组作为Request原语声明.

Policy原语定义了策略规则的语义.该字段制定了PML策略的解释器应如何解释策略规则.为简单起见,本文采用逗号分隔值(comma-separated values, CSV)来作为策略规则的语法.与Request原语类似,经典的Policy原语也由主体、客体、动作三元组构成:  
p = sub, obj, act  
与其相应的策略规则可以为:  
p, alice, data1, read  
策略中的每一行都是一条策略规则.策略规则的第一个元素是策略类型,该字段与Policy原语相对应,通过指定不同的策略类型可以支持多个Policy原语,如p, p2等.通常情况下,不需要指定多个策略类型.上面的策略规则与之前的经典Policy原语的绑定关系为:  
(alice, data1, read) à (p.sub, p.obj, p.act)  
上述绑定类似于变量赋值,可以在Matcher原语中使用.  
16:33:29  
肉食大灰兔o(╯□╰) 2017/12/11 16:33:29  
  
Matcher原语定义了策略规则如何与访问请求进行匹配的匹配器,其本质上是布尔表达式.可以理解为Request、Policy等原语定义了关于策略和请求的变量,然后将这些变量代入Matcher原语中求值,从而进行策略决策.Matcher原语支持+, -, \*, /等算数运算符、==, !=, >, <等关系运算符以及&& (与), || (或), ! (非)等逻辑运算符.Matcher原语最终的运算结果应为布尔值0或1,1表示策略规则满足该匹配器,0则表示不满足.与上节中列举的经典Policy原语相对应的Matcher原语为:  
m = r.sub == p.sub && r.obj == p.obj && r.act == p.act  
该Matcher原语表示,访问请求中的主体、客体、动作三元组应与策略规则中的主体、客体、动作三元组分别对应相同.  
肉食大灰兔o(╯□╰) 2017/12/11 16:33:34  
  
Effect原语定义了当多个策略规则同时匹配访问请求时,该如何对多个决策结果进行集成以实现统一决策,决策结果包括allow和deny两种.Effect原语支持some, any等量词以及条件关键字where、&& (与), || (或), ! (非)等逻辑运算符.some量词判断是否存在一条策略规则满足匹配器,any量词则判断是否所有的策略规则都满足匹配器.Where关键字表示后面跟策略规则的筛选条件.Effect原语可以表达允许优先(allow-override)、拒绝优先(deny-override)等多种决策集成方法.允许优先可以表达为:  
e = some(where (p.eft == allow))  
该Effect原语表示如果存在任意一个决策结果为allow的匹配规则,则最终决策结果为allow,即allow-override.其中p.eft表示策略规则的决策结果,可以为allow或者deny.当不指定规则的决策结果时,取默认值allow.利用Effect原语还可以表达拒绝优先:  
e = !some(where (p.eft == deny))  
该Effect原语表示不存在任何决策结果为deny的匹配规则,则最终决策结果为allow,否则为deny.Effect原语还可以利用逻辑运算符进行连接:  
e = some(where (p.eft == allow)) && !some(where (p.eft == deny))  
该Effect原语表示当至少存在一个决策结果为allow的匹配规则,且不存在决策结果为deny的匹配规则时,则最终决策结果为allow.这时正向授权和负向授权同时存在,并且拒绝优先.  
16:40:06  
广州qingo 2017/12/11 16:40:06  
  
这个中文说明放在哪里啊  
水无 2017/12/11 16:40:07  
  
和实际的访问怎么对应起来？  
水无 2017/12/11 16:40:23  
  
我用了mysql库，就一个表  
水无 2017/12/11 16:40:37  
  
水无方  
如果前端菜单是从后台根据不同用户返回，那这个菜单定义为obj还是act？  
肉食大灰兔o(╯□╰) 2017/12/11 16:57:37  
  
obj啊  
2017/12/11 17:01:26  
水无 2017/12/11 17:01:26  
  
这个casbin感觉很抽象  
肉食大灰兔o(╯□╰) 2017/12/11 17:02:11  
  
casbin的表达能力几乎可以达到图灵机

肉食大灰兔o(╯□╰)o  
不用function是SQL的表达能力，用function就是图灵机的表达能力  
肉食大灰兔o(╯□╰) 2017/12/11 17:02:42  
  
所以有些抽象  
肉食大灰兔o(╯□╰) 2017/12/11 17:03:17  
  
可以看一下几个经典的例子，ACL, RBAC, ABAC之类的，体会一下

https://github.com/CyberlifeCN/cyber-auth-api/blob/master/conf/authz\_policy.csv