ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.14, No.28,October . 2018
E-mail: jslt@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

风险自适应访问控制算法研究

朱兴萍

(盐城幼儿师范高等专科学校,江苏 盐城 224005)

摘要:因开放环境的日益复杂性、异构性和动态性,对当前的访问控制提出了独特的安全挑战。详细分析了RAdAC中的安全风险测量(SRM)算法核心以及最终访问决定(FAD)策略冲突检测,并通过对当前常用访问控制的对比分析,提出了以后进一步的研究内容,为该领域的研究提供了思路。

关键词:访问控制;风险自适应访问控制;安全风险测量;最终访问决定;策略冲突检测

中图分类号:TP309.2 文献标识码:A 文章编号:1009-3044(2018)28-0012-03

DOI:10.14004/j.cnki.ckt.2018.3249

1 引言

风险自适应访问控制(RAdAC,Risk-Adaptable Access Control)是美国国家安全局(NSA)研究的下一代动态访问控制方法,RAdAC能够根据当前平台状态以及特殊情况灵活动态的对用户进行授权,给用户提供最严格的访问策略,提供多级安全(MLS,Multi Level Security)级别的访问。基于上述原因,提出一种基于RAdAC的访问控制方法,使其具备动态分析安全风险和操作需求的能力,以适应实时环境和复杂情况。

2 RAdAC特征分析

RAdAC是一种基于规则的动态访问控制策略,在使用中,实时评估用户操作需求并计算授权访问风险。RAdAC包含以下三部分[1]:安全风险测量(SRM,Security Risk Measurement)、操作需求测定(OND,Operational Need Determination)以及最终访问决定(FAD,Final Access Decision)。

安全风险测量(SRM):SRM会根据用户操作的不同、用户状态、环境因素、上下文、访问目标特征等多种特性通过阈值判定、加权等计算方式得出本次访问风险值。与传统结果不同的是,最后的风险值并不是大、小或"True""False"等二进制值,而是一个范围,可能是30%-50%或者风险中等和风险大之间的状态。同时SRM还需具备一定的机器自学能力,如果用户在操作中多次访问核心数据资源,则必须提高风险等级[2]。

操作需求测定(OND):OND使得用户在特殊情况下能够超越风险必须访问某资源的一种判定。传统方式下,用户属于一个部门或者拥有某几个角色,能够访问的资源都是固定的,但是在紧急情况或者风险较大的时候,用户可能需要拥有超过自身权限的访问能力。系统需要用户权限和访问要求做交互来得到最后的操作需求判定。

最终访问决定(FAD):FAD根据SRM和OND的结果,通过访问决定函数(ADF,Access Decision Function)来做出最终结果。由于用户每次操作的风险值和需求都是变化的,因此在多级安全的系统中,用户的最终访问结果并不是固定的。FAD的结果是二进制值,允许或拒绝。

3 RAdAC判定方法

根据对RAdAC模型的分析,RAdAC需要对SRM和OND各属性进行分类管理,并根据属性值和相应的评估函数进行计算,得到FAD。图1给出了RAdAC算法流程图[3]:

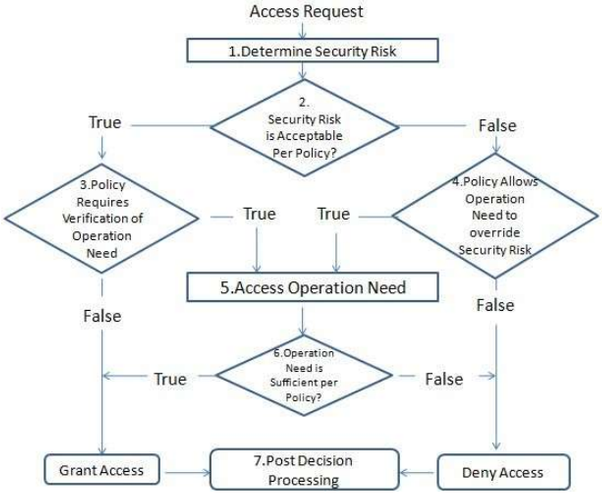

图1 RAdAC算法流程图

3.1 SRM模块

3.1.1 系统性能评估

定义1:系统安全态势评估模型中的性能信息P用$(t,\gamma,\mu,k,\rho,\delta,\theta)$表示,其中$t$是性能产生的时间;$\gamma$是系统CPU使用率;$\mu$是系统内存使用率;$k$是数据库连接数;$\rho$是流量;$\delta$是丢包率;$\theta$代表磁盘I/O读写状态。在$t$时刻,其性能参数$(\gamma,\mu,\kappa,\rho,\delta,\theta)$的最小值都为0,对应的最大值为$(\gamma,\mu,\kappa_{max},\rho_{max},1,\theta)$,其中$\kappa_{max}$是最大允许连接数;$\rho_{max}$是最大流量。系统性能由当前可利用资源来衡量,采用如下公式计算

收稿日期:2018-08-25

作者简介:朱兴萍(1982—),女,江苏滨海人,讲师,主要研究方向为数学教育。

节点当前性能值P：

$$P = 1 - (\gamma \times w_1 + \mu \times w_2 + \frac{\kappa}{\kappa_0} \times w_3 + \frac{\rho}{\rho_0} \times w_4 + \delta \times w_5 + \theta \times w_6) \quad (1)$$

其中，$w_1, w_2, w_3, w_4, w_5, w_6$为各性能参数的权重。设在某时间段开始时刻，某系统的性能参数为$P_1(\gamma_1, \mu_1, \kappa_1, \rho_1, \delta_1, \theta_1)$，该时间段结束时刻的性能参数为$P_2(\gamma_2, \mu_2, \kappa_2, \rho_2, \delta_2, \theta_2)$，则性能变化量

$$\Delta P = P_1 - P_2 = (\gamma_2 - \lambda_1) \times w_1 + (\mu_2 - \mu_1) \times w_2 + (\frac{\kappa_2 - \kappa_1}{\kappa_0}) \times w + (\frac{\rho_2 - \rho_1}{\rho_0}) \times w_4 + (\delta_2 - \delta_1) \times w_5 + (\theta_2 - \theta_1) \times w_6 \quad (2)$$

使用性能变化量ΔP仅代表理论安全性能变化，通常需要根据经验值η进行修正，计算公式为：

$$Security = (1 - \eta) \times Fun + \eta \times \Delta P \quad (3)$$

其中η为评估函数Fun的修正系数，取值为$[0,1]$，当η取值越小时，说明ΔP越能反映安全状况，理论和实际结果对应；反之，则说明误差较大。

3.1.2 历史记录分析

本节采用Holt-Winters模型把具有线性趋势、季节变动和随机波动的时间序列进行分解研究，并与指数平滑法(Exponential Smoothing)相结合，分别对长期趋势、趋势的增量和季节波动做出估计，然后建立预测模型，得到预测值。该模型由以下3个方程和一个预测公式组成：

$$\begin{aligned} S_t &= \alpha X_t / I_{t-L} + (1 - \beta)(S_{t-1} + b_{t-1}) \\ b_t &= \gamma(S_t - S_{t-1}) + (1 - \lambda) b_{t-1} \\ I_t &= \beta X_t / S_t + (1 - \beta) I_{t-L} \\ f_{t+m} &= (S_t + b_t m) I_{t+m-L} \end{aligned} \quad (4)$$

其中X_t为时间序列，L为季节长度，S是稳定成分，b是线性趋势成分，I为季节成分，α, β, γ为加权系数，取值在[0,1]之间，可用MAD,MSE或MAPE等方法选取。预测公式为f_{t+m}。

为了初始化，设置

$$S_t = \frac{1}{L}(X_1 + X_2 + ... + X_L) \quad (5)$$

设置b_t在$L + k$个时间范围内，$b_t = \frac{1}{k}(\frac{X_{L+1} - X_1}{L} + \frac{X_{L+2} - X_2}{L} + ... + \frac{X_{L+k} - X_k}{L})$ (6)

如果k足够长，当$L = k$时，则为两个完整的周期。

3.1.3 资源重要性判定

通常我们认为，资源越重要，提供服务越高级，则访问风险越高：

$$Q(req_i, res_j) = S_t(1 - \Gamma(req_i, res_j, t - 1)) \quad (7)$$

其中，S_t表示请求资源的重要性，访问风险$Q(req_i, res_j)$与S_t成正比。$\Gamma(req_i, res_j, t - 1)$代表$req_i$对$res_j$在离时间$t$最近一个时间戳内的信用评价。信任度越高，访问请求风险越小。

3.1.4 访问路径

在多级安全的分布式系统中，当用户进行跨域访问时，不同子系统所在的层级不同，信任等级也不同。因此，越靠近的子系统，信任程度越高，即访问风险越低。子系统的信任程度与访问者的请求距离(等级)进行加权。

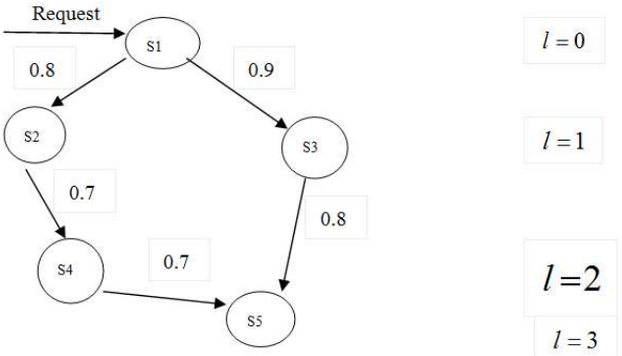

图2 访问者路径

在图2中，当访问者请求资源服务时，从子系统$s1$跨域登录到$s2$时，也就是$l = 1$，信任程度$t = 1 \times 0.8 = 0.8$；当从$s2$再登录到子系统$s4$时，即$l = 2$，信任程度$t = 1 \times 0.8 \times 0.7 = 0.56$；当从$s4$登录到$s5$时，即$l = 3$，信任程度$t = 1 \times 0.8 \times 0.7 \times 0.7 = 0.392$。

定义2：设$\{W_1, W_2, ..., W_k\}$为分布式系统中的子系统，$D(req_i, res_k)$表示第k个子系统资源对访问者i的直接信任，则访问风险为：

$$R(req_i, res_j) = 1 - \sum_{k=1}^{L}(\rho(W_k) \times D(req_i, res_k)) \quad (8)$$

其中L为子系统的路径，$\rho(W_k)$为路径加权函数：

$$\rho(W_k) = \prod_{d=0}^{l} L(res_i, res_j), l \geq 1 \quad (9)$$

其中$L(x_i, x_j)$表示访问者从子系统i登录到子系统j信任程度的降低。

3.1.5 综合计算

系统在为请求者服务时，既包括合法用户的正常请求服务，也包括系统异常和遭受攻击时的资源服务。对风险的判读既要偏重理论风险的计算，也要关心系统性能的状态，并根据历史记录预测风险趋势。图3给出了SRM执行流程图。

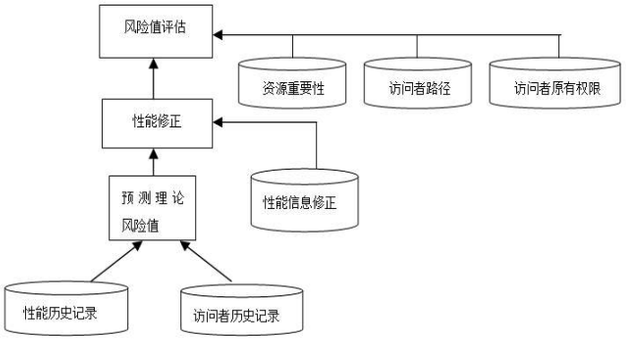

图3 SRM执行流程图

Step 1：利用访问历史记录，预测理论风险值。

Step 2：获取系统安全属性信息通过性能修正算法对预测评估结果进行修正。如果风险较大，则直接给出风险结果。否则进入Step 3。

Step 3：对预测值、系统性能值、资源重要性、访问者路径和

访问者原有权限进行加权，这里采取专家意见法决定权重。

3.2 ADF平台策略冲突检测

ADF需要综合考虑SRM和OND因素，甚至需要考虑额外可能引起特殊情况的上下文信息。因此，ADF需要判定SRM与OND的优先级，在此扩充文献[4]的规则冲突规则。

定义3：SRM风险结果{Trust，Midtrust，LowTrust，Untrust}，风险程度逐渐增加。当风险结果为Untrust时，则访问风险极高，访问者完全不可信任。

规则1：ADF策略冲突检测规则：SRM评估函数和OND不采取加权方式，也不是简单的OND覆盖SRM关系。当SRM风险值很大时，SRM评估结果覆盖OND。反之，当SRM处在可接受范围内时，如果OND重要，则结果为Permit，否则Deny。具体如下：

ADF访问策略判定函数优先级：

ADF∈{AllPermit，SRMOverridePermit ，SRMOverrideDeny，ONDOverridePermit，ONDOverrideDeny，AllDeny }

AllPermit：风险结果为Trust或Midtrust且操作必须时采取的安全策略，此时结果为Permit。

SRMOverridePermit：风险结果为Trust或Midtrust，并且系统认为无须做操作需求交互，则结果为Permit，即SRM Permit优先原则。

SRMOverrideDeny：风险结果为Untrust时，即使用户有操作需求以及正常情况下的权限，系统也必须拒绝访问，即SRM Deny优先原则。

ONDOverridePermit：当风险结果为LowTrust时，如果在紧急情况下有操作需求，系统与用户正常访问权限及访问需求进行交互，认为可以适当放宽该用户权限。此时满足OND Permit允许覆盖原则。

ONDOverrideDeny：如果对OND评估结果为该用户不需要对资源进行访问，即使无访问风险，用户也缺乏操作权限，结果为Deny。

AllDeny：系统评估结果认为该用户对资源的访问既无必要，访问风险也处在Untrust或LowTrust，则结果为Deny。

4 RAdAC算法分析比较

本节通过对RAdAC算法和当前RBAC、PBAC和ABAC算法进行比较，总结出RAdAC算法特性：

（1）平台扩展性对比:RAdAC可以在数据库或XACML的基础上，增加对平台安全性、用户属性的判定，依据相应的判定函数实施访问判决，克服了传统方式下的缺陷。

表1 平台扩展性对比

Platform Security Attribute	Policy Determine	Role Determine	Attribute Determine	Platform Heuristic	Dynamic Attribute	Risk Determine
RBAC	√	√				
PBAC	√	√	√		√	
ABAC			√		√	
RAdAC	√	√	√	√	√	√

（2）策略合成算法：目前大部分平台安全策略工作采用XACML提供的集中式策略合成算法，为了数据的安全访问，往往直接遵守最小特权原则（deny override），无法根据具体情况作出不同判断。RAdAC根据当前风险等级和操作需求，灵活确定资源访问权限，在保证数据安全的同时，尽可能满足访问需求。

（3）RAdAC算法执行效率：RAdAC在访问控制中，需要遍历和检验SRM中不同属性的度量值，同时平台需要与用户交互确定OND结果，并结合SRM进行ADF判定。因此，SRM属性值的选取和OND的交互是RAdAC算法执行速度的关键。假设分别对N个资源发出请求，平台环境安全属性及状态个数为M个，则ADF策略匹配复杂度为$\theta(N\times M\times 6)$。对于不同安全级别的平台系统，可以收集不同数量的属性值和不同等级的OND交互，以提高执行效率。

5 结束语

RAdAC是下一代的访问控制机制，用以解决目前常用的访问控制机制在平台安全属性和安全策略方面的不足。本文首先对RAdAC结构和执行流程进行了分析，制定了算法规则和属性定义，并对RAdAC算法与常用访问控制方法进行了对比分析。由于涉及到大量的动态属性和启发式算法，因此下一步的工作重点是进一步扩展SRM属性判定方法和ADF冲突检测类型，使其更具有工程实际意义。

参考文献：

[1] Rahim Choudhary. A Policy Based Architecture for NSA RAdAC Model[R].USA ,New York,2005.

[2] Machon Gregory,Peter Loscocco.Using the Flask Security Architecture to Facilitate Risk Adaptable Access [EB/OL].http://www.nsa.gov/research/_files/selinux/papers/radac07-paper.pdf

[3] Mcgraw, R. W. Risk-Adaptable Access Control(RAdAC) [EB/OL]. http://csrc. nist. gov / news events / privilege-management-workshop/ radac-Paper0001.pdf

[4] Time series Forecasting using Holt-Winters Exponential Smoothing. http://www. it. iitb. ac. in/~praj / acads / seminar / 04329008_ExponentialSmoothing.pdf

【通联编辑：代影】