张玉秀, 张昊. 基于冗余消除和 Chameleon 聚类的分布式 XACML 模型[J]. 智能计算机与应用,2026,16(6):97-102. DOI: 10.20169/j.issn.2095-2163.24101301

基于冗余消除和 Chameleon 聚类的分布式 XACML 模型

张玉秀, 张 昊
(国网上海市电力公司信息通信公司, 上海 200072)

摘 要: 针对访问控制策略评估效率面临的严峻挑战,提出一种新的分布式 XACML 策略评估模型(RCD-XACML),使用冗余消除和聚类技术消除单个策略决策点评估性能的局限性。将策略划分为多个不相交的原子规则,进行规则冗余消除,缩减策略集;从属性加权的角度定义规则相似性,通过 Chameleon 多阶段层次聚类算法对策略中的规则进行聚类;每个访问请求基于相似度快速定位最可能适用的规则簇,从而减少处理访问请求所需的规则数量。本文所提出的 RCD-XACML 与现行的 Sun-XACML 以及 SBA-XACML 引擎相比,评估性能显著提升。

关键词: 访问控制; XACML; 冗余消除; Chameleon 聚类; 策略评估
中图分类号: TP18 **文献标志码**: A **文章编号**: 2095-2163(2026)06-0097-06

A distributed XACML model based on redundancy elimination and Chameleon clustering

ZHANG Yuxiu, ZHANG Hao
(Information and Communication Company, State Grid Shanghai Municipal Electric Power Company, Shanghai 200072, China)

Abstract: In response to the extremely severe challenges faced by access control policy evaluation efficiency, a brand new distributed XACML policy evaluation model known as RCD-XACML is put forward. This innovative model makes use of redundancy elimination and clustering techniques to overcome the limitations in the evaluation performance of a single policy decision point. The policy is divided into numerous disjoint atomic rules, and then rule redundancy is eliminated in order to shrink the policy set. Rule similarity is defined from the perspective of attribute weighting. The Chameleon multi-stage hierarchical clustering algorithm is employed to cluster the rules within the policy. For each access request, the most likely applicable rule cluster can be quickly located based on similarity. As a result, this reduces the number of rules required to handle access requests. When compared with the currently existing Sun-XACML and SBA-XACML engines, the RCD-XACML proposed in this paper demonstrates a significant improvement in evaluation performance. This enhanced performance can lead to more efficient access control management and better protection of sensitive resources.

Key words: access control; XACML; redundancy elimination; Chameleon clustering; policy evaluation

0 引 言

在现代互联网环境下,访问控制的应用不断扩大,云计算、物联网等快速发展的技术都将访问控制作为确保系统安全的重要方法。基于属性的访问控制(ABAC)具有 XACML(eXtensible Access Control Markup Language)框架的支持,该框架创建策略和规则,并动态计算发出访问请求的用户属性是否满足策略与规则的属性来进行授权决策,可以解决资源保护所面临的细粒度问题和动态性权限控制。授权服务系统的运行效率与 XACML 的评估性能密切相关,故要求 XACML 具有高效的评估效率[1-4]。然而,随着系统用户与资源的急剧增长,策略与规则的数量与复杂性都大大增加,XACML 评估性能改进出现以下瓶颈:

(1)策略集冗余导致策略评估速度以及精确度降低。

(2)在成千上万条策略中蛮力检索到有效策略的概率非常低,评估时延、系统开销不容忽视。

(3)所有访问请求都必须发送到一个单独的策略

作者简介: 张玉秀(1997—),女,硕士。主要研究方向:访问控制,数字化项目建设管理。Email:1004992567@qq.com; 张 昊(1997—),男,硕士。主要研究方向:数据治理,数据分析及应用。

收稿日期: 2024-10-13

决策点(PDP),无法并发处理多个服务器上的请求。

因此,亟需设计一种高性能的访问控制策略评估模型。目前,学者们从不同角度研究了策略评估优化方法:在不修改策略集的情况下提出全新的评估过程[5-8];或保持评估过程不变,通过修改规则、设置规则索引、建立规则缓存机制以及构造决策图等方法提高策略评估性能[9-23]。由于信息系统日益多样化,实际的访问控制场景的要求越来越复杂,当前策略评估方法的效率仍有待提高。针对以上问题,本文设计了一种分布式 XACML 框架(RCD-XACML),其中包括负责策略集预处理过程的策略管理点(RCPAP),以及分布式策略决策点体系结构(DPDP),主要贡献如下。

(1)在不影响策略评估结果的前提下,根据规则评估结合算法消除冗余规则,并提出将策略和规则原子化,进一步消除规则间的冗余状态,减小冗余消除的粒度,实现策略评估的前期优化。

(2)从属性加权的角度定义规则相似性,通过 Chameleon 多阶段层次聚类算法基于规则相似性将策略集精确地聚类为多个簇。此过程只需考虑策略集中规则的主要属性,而不需要考虑策略集的内在特征。

(3)提出分布式 XACML 评估模型,通过执行聚类,多个策略决策点(PDP)可同时独立工作,并发处理多个服务器上的请求,每个 PDP 仅处理适用于加载到其中的策略集的请求,极大程度上缩减了评估的规则量,从而消除了传统 XACML 评估模型中单个 PDP 计算性能的限制。

1 分布式策略评估模型 RCD-XACML

根据 XACML 标准数据流模型,本文提出的 RCD-XACML 实现了完整的访问控制系统,并使用 RCPAP 和 DPDP 分别代替传统系统中的策略管理点(PAP)和策略决策点(PDP)。RCD-XACML 框架的数据流模型如图 1 所示。

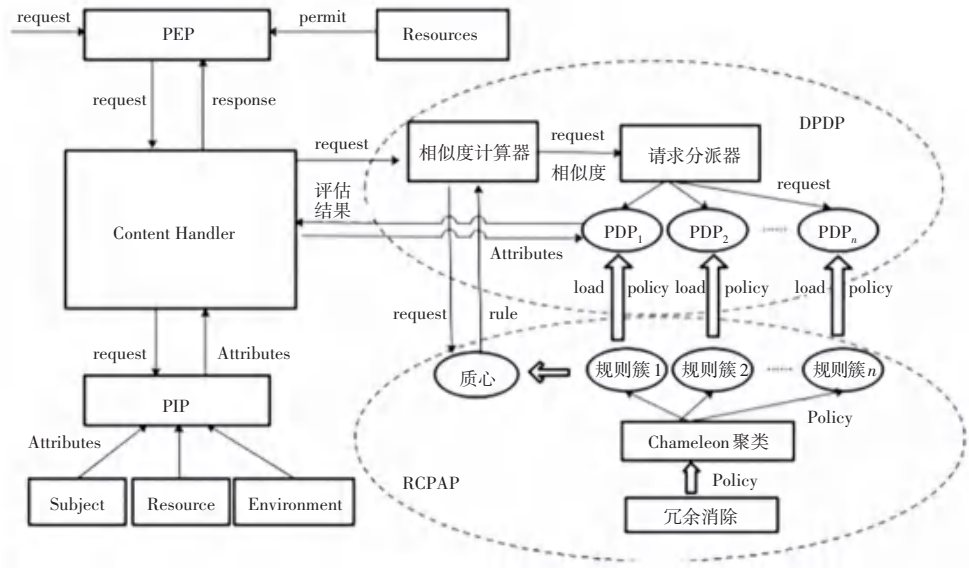


图 1 RCD-XACML 框架的数据流模型

Fig. 1 Data flow model of RCD-XACML framework

RCPAP 存储并维护由管理员制定的安全策略,负责策略集中规则的冗余分析以及聚类。冗余消除组件将复合规则转换为原子规则,根据规则组合算法检测并删除不影响实际判断的冗余规则,消除规则之间的冗余状态;其次,Chameleon 聚类组件将策略中的规则聚类到不同的簇中,并结合分布式 PDP 体系结构,将规则簇分别加载到各个 PDP 中,减少单个 PDP 中的策略量,以便分布式 PDP 可以独立工作。同时,所有簇的质心都存储在单独的组件中,实现策略匹配的前期优化。

DPDP 是 RCD-XACML 的核心组件,包括相似度计算器、请求分派器以及多个 PDP 组件。其中,相似度计算器负责计算访问请求与所有规则簇质心的相似度,得到的最大相似度表示该质心所在的规则簇最有可能适用于当前请求。基于最大相似度,请求分派器将请求发送到规则簇所在的 PDP。PDP 评估请求和规则以获得结果,并将结果返回到上下文处理程序。如果评估结果不适用,则使用最可能适用的第 2 个规则簇,依此类推。由于加载到 PDP 的策略数量相对较小,而不是整个策略集,因此在提

高效率方面具有显著差异。

2　规则冗余消除

虽然策略规则冗余并不影响访问行为决策，但规则的数量影响评估效率，会降低评估精确度。此章节主要描述规则间的冗余状态消除方法。

规则状态覆盖时，两规则可能同时适用于某一请求，根据 XACML 提供的组合算法，本文借鉴文献[10]中总结的冗余规则判定定理，证明过程不再重复。

定理 1　若规则组合算法为 Permit-overrides，R_j 的状态覆盖 R_i 的状态：

(1)若 R_j. Effect = Permit，则 R_i 是冗余规则。

(2)若 R_j. Effect = Deny，则当 R_i. Effect = Deny 时，R_i 是冗余规则；当 R_i. Effect = Permit 时，R_i 不是冗余规则。

定理 2　若规则组合算法为 Deny-overrides，R_j 的状态覆盖 R_i 的状态：

(1)若 R_j. Effect = Deny，则 R_i 是冗余规则。

(2)若 R_j. Effect = Permit，则当 R_i. Effect = Permit 时，R_i 是冗余规则；R_i. Effect = Deny 时，R_i 不是冗余规则。

规则状态相交时，无法使用状态覆盖下的冗余判定方法消除冗余规则。由于包含原子属性的规则更易于存储和处理，因此本文提出将复合规则分解为原子规则，减小冗余消除粒度，从而消除规则间的冗余状态。

定义 1　原子属性　如果规则中的属性没有超过一个可能缺少的元素，则该属性称为原子属性。

定义 2　原子规则　如果一个规则的所有属性都是原子属性，则称该规则为原子规则。

定义 3　复合属性　如果规则中的一个属性有多个元素，则该属性称为复合属性。

定义 4　复合规则　如果一个规则中含有复合属性，则称该规则为复合规则。

本文将复合属性划分为多个原子属性，并与其他属性的元素进行复合，最终得到只包含原子属性的原子规则，为冗余状态消除的理论分析奠定基础。例如，策略集合 P_1，其规则组合算法为 Deny-Overrides，规则 R_1：

Sub ∈ {Bob, John} ∧ Res ∈ {file1, file2} ∧ Act = read ∧ Cond = All⇒Permit 可将其分解为如下原子规则：

(1) R_{11}：Sub = Bob ∧ Res = file1 ∧ Act = read ∧ Cond = All⇒Permit

(2) R_{12}：Sub = Bob ∧ Res = file2 ∧ Act = read ∧ Cond = All⇒Permit

(3) R_{13}：Sub = John ∧ Res = file1 ∧ Act = read ∧ Cond = All⇒Permit

(4) R_{14}：Sub = John ∧ Res = file2 ∧ Act = read ∧ Cond = All⇒Permit

对于规则 R_2：

Sub ∈ {Bob, Mark} ∧ Res ∈ {file1, file2} ∧ Act = read ∧ Cond = All⇒Deny 可将其分解为如下原子规则：

(1) R_{21}：Sub = Bob ∧ Res = file1 ∧ Act = read ∧ Cond = All⇒Deny

(2) R_{22}：Sub = Bob ∧ Res = file2 ∧ Act = read ∧ Cond = All⇒Deny

(3) R_{23}：Sub = Mark ∧ Res = file1 ∧ Act = read ∧ Cond = All⇒Deny

(4) R_{24}：Sub = Mark ∧ Res = file2 ∧ Act = read ∧ Cond = All⇒Deny

通过上述方法可将规则粒度降至最低，根据规则状态定义可知，R_{21} 的状态覆盖 R_{11} 的状态，规则组合算法为 Deny-Overrides，R_{21}. Effect = Deny，根据定理 2 推断出 R_{11} 为冗余规则。同理可得 R_{12} 为冗余规则，将其删除后还剩 R_{13}，R_{14}，R_{21}，R_{22}，R_{23}，R_{24} 共 6 个规则。

3　规则 Chameleon 聚类

3.1　规则相似度

在消除规则冗余的前提下，本文提出规则相似度的概念，将规则进行聚类。

策略决策点通过比较规则与请求的属性进行评估，因此若 2 个规则的 Subject、Resource 或 Action 的值相同，并且为请求属性指定了相似的条件时，则这 2 个规则很大概率同时适用于同一请求。将相似的规则聚集在一起，请求只需与可能适用的规则簇进行匹配，而忽略不适用的规则。2 个规则相似度的定义公式为：

$$\text{simRule} = w_1 \times \text{simSub} + w_2 \times \text{simAct} + w_3 \times \text{simRes} + w_4 \times \text{simCond} \quad (1)$$

其中，simSub，simAct，simRes，simCond 分别表示主体、操作、资源和条件属性的相似度。

当 2 个规则中同一属性的属性值相等时，属性相似度取值 1。否则，属性相似度取值 0。例如，R_{21}，R_{23} 的 Subject 值分别是 Bob，John。因此 simSub

取值 0。由于在匹配请求与规则的过程中,按顺序比较属性,首先比较请求和规则中的主体属性,其次比较操作、资源和条件属性。若请求和规则中的主体不同,则规则不适用,不再进行比较操作、资源和条件属性。因此前比较属性的权重应该大于后比较属性的权重,且权重值应满足约束条件如下:

$$w_1 + w_2 + w_3 + w_4 = 1 \quad (2)$$

本文取 $w_1 = 0.4, w_2 = 0.3, w_3 = 0.2, w_4 = 0.1$。以 R_{13}, R_{14}, R_{21} 三条规则为例:

$\text{simRule}(R_{13}, R_{14}) = 0.4 \times 1 + 0.3 \times 1 + 0.2 \times 0 + 0.1 \times 1 = 0.8$

$\text{simRule}(R_{13}, R_{21}) = 0.4 \times 0 + 0.3 \times 1 + 0.2 \times 1 + 0.1 \times 1 = 0.6$

$\text{simRule}(R_{14}, R_{21}) = 0.4 \times 0 + 0.3 \times 1 + 0.2 \times 0 + 0.1 \times 1 = 0.4$

3.2 聚类算法

Chameleon 聚类算法同时考虑簇中规则之间的互连性(RI)以及邻近性(RC)来确定相似度,克服了其他聚类方法的弱点。因此,本文中采用了 Chameleon 聚类算法划分策略集[24], G_k 图中的每个顶点表示策略中的一个规则,连接顶点的边的权值为规则的相似度、即在 3.1 节中计算出的 $\text{simRule}(R_i, R_j)$, 相似度越大,距离越小,边的权值越大。

3.3 质心

为了确定访问请求最可能适用的规则簇,本文求出每个规则簇的质心。质心是与簇中的所有规则都具有最大相似性的特定规则,其属性值为规则簇中每个属性出现频率最高的值。

4 仿真实验

4.1 实验环境和数据集

4.1.1 实验环境

本文在一台运行 macOS Catalina 系统的便携式计算机上进行实验。由于 Sun-XACML 和 SBA-XACML 都是使用 Java 实现,为比较 RCD-XACML 与其之间的性能差异,本文同样使用 Java 实现了 RCD-XACML,从而避免编码语言对评估性能的影响。

4.1.2 策略集

为了模拟实际的应用场景,本文选择了图书馆管理系统(LMS)、虚拟会议系统(VMS)以及拍卖销售管理系统(ASMS)中的 XACML 访问控制策略。LMS、VMS、ASMS 分别含有 945、720、1 760 条规则。考虑到当前应用系统中使用的规则规模显著扩展,有必要扩充数据集。通过解析策略集得到规则中属性的所有值,根据不同的主体、动作、资源和条件的笛卡尔积,构造新的规则添加到原始策略中。最终,VMS、LMS 和 ASMS 策略中的规则数分别扩展到 3 000、6 000 和 9 000。

4.1.3 请求集

通过分析策略集得到规则中每个属性的所有值,这些值构成规则中每个特定属性的域。本文采用随机分配的方法,从每个属性域中随机选取一个值,并将这些值组合成每个请求。共构造了 5 个请求集,分别包括 2 000,4 000,6 000,8 000,10 000 个请求。

4.2 性能测试和比较

4.2.1 预处理结果

Chameleon 聚类生成的规则簇数量会影响评估效率。若簇数量较少,每个簇中的规则会较多,策略决策点可能会评估大部分的规则,从而影响评估效率。若簇的数量与规则数量相同,每个簇都有一个质心用来决定与请求相似度最高的簇,则计算访问请求与质心相似度的过程相当于蛮力搜索所有规则,因此规则簇数量太多,亦不合理。本文聚类结果如图 2 所示,避免簇的数量过多或过少,在实施 Chameleon 算法后,VMA、LMS、ASMS 三个策略集中的规则分别分为 20、47、68 个簇,每个规则簇平均有 130~150 条规则。

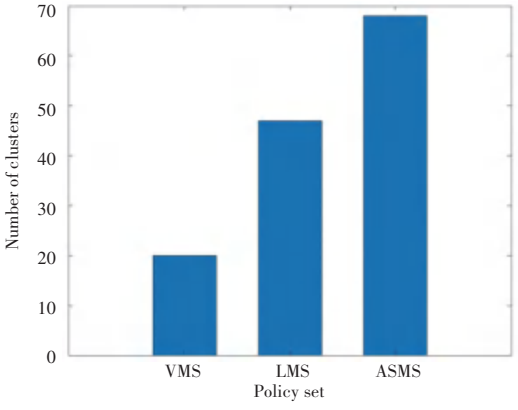


图 2 聚类结果

Fig. 2 Clustering results

4.2.2 评估时间的比较

为了全面评估 RCD-XACML 在响应访问请求方面的性能,并找出 RCD-XACML 与其他策略评估引擎相比的确切改进。本文则在 LMS、VMS、ASMS 三个策略集上实现了 5 个评估方法,分别是 Sun-XACML、SBA-XACML、不加聚类的 R-XACML、不加冗余消除的 CD-XACML 以及 RCD-XACML(优化方法结合),可以分别说明 2 个技术对评估性能的影响。

由于授权处理时间是性能的关键指标，因此本文测试比较评估时间。针对每种评估方法分别进行 3 个策略集的 10 次评估测试，并计算每种评估方法的平均时间作为最终结果，对比结果如图 3～图 5 所示。3 种评估方法在 3 个策略集上评估 10 000 个访问请求的平均时间见表 1。

表 1　10 000 个访问请求的评估时间和提升率

Table 1　Evaluation time and promotion rate of 10 000 access requests

策略集	规则数	评估时间/ms					本文方案提升率/%	
		Sun-XACML	SBA-XACML	R-XACML	CD-XACML	RCD-XACML	Sun-XACML	SBA-XACML
VMS	3 000	40 735	7 128	26 578	50	34	1 197	209
LMS	6 000	70 328	14 258	35 164	67	44	1 597	323
ASMS	9 000	92 773	21 387	46 386	70	48	1 932	445

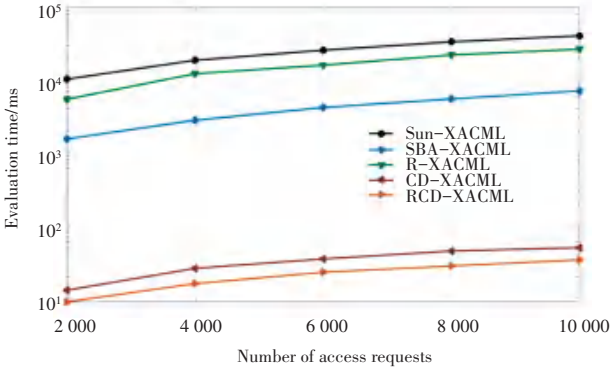


图 3　VMS 策略集评估时间对比

Fig. 3　Comparison of evaluation time of VMS policy set

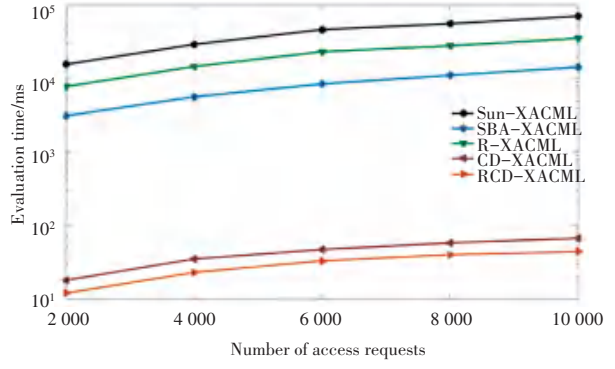


图 4　LMS 策略集评估时间对比

Fig. 4　Comparison of evaluation time of LMS policy set

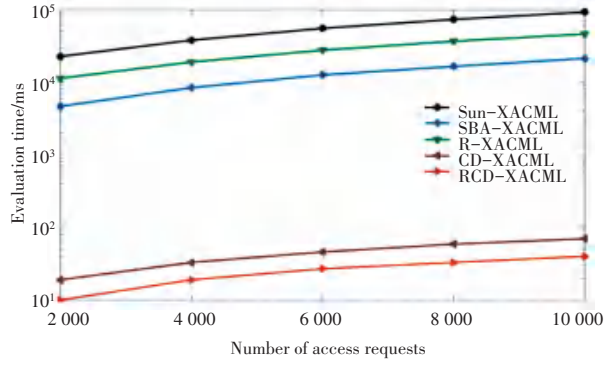


图 5　ASMS 策略集评估时间对比

Fig. 5　Comparison of evaluation time of ASMS policy set

由此可以看出：

(1) 随着请求数量的增加，Sun－XACML，R－XACML，SBA－XACML，CD－XACML，RCD－XACML 的平均评估时间会增加。

(2) 在消除冗余规则以及规则间的冗余状态后，Sun-XACML 的评估时间平均是 R-XACML 的评估时间 1.8 倍，表明只消除冗余规则评估效率提高得相对较少。

(3) CD-XACML 的评估效率比 Sun-XACML 的评估效率平均提高 15.8 倍，比 SBA-XACML 的评估效率平均提高 3.26 倍，说明基于聚类技术的分布式 PDP 对评估性能的改进非常显著。

(4) RCD－XACML 评估效率比 CD－XACML 的评估效率平均提高 1.48 倍，即结合冗余消除和聚类技术的 RCD-XACML 评估性能得到进一步提高。

4.2.3　评估规则数量的比较

通过实现分布式 PDP 模型，访问请求与其最可能适用的规则进行匹配，如果评估结果不适用，则使用第 2 个最可能适用的规则簇，依此类推。为了进一步验证该方法的有效性，本文在 Sun－XACML、RCD－XACML 上分别使用 3 个策略集处理 10 000 个访问请求，比较评估一个请求的平均规则数量。对于每种评估方法，分别对每个策略集进行 10 次评估测试。本文选择 2 种方法评估的平均规则数量作为最终结果并绘制图 6。

由图 6 可知，随着策略集中策略数量的增加，Sun-XACML 评估一个请求需要匹配的规则数大约满足线性增长。而 RCD-XACML 评估一个请求需要匹配的规则数稳定在 100 左右。例如，在 VMS 策略集中，Sun-XACML 需评估 1 561 条规则，而 RCD-XACML 仅需 112 条规则。显然，使用 RCD-XACML 的确减少了评估的规则数量。该实验证明，通过将最可能适用的规则聚集在一起并优先评估，可提高评估效率。

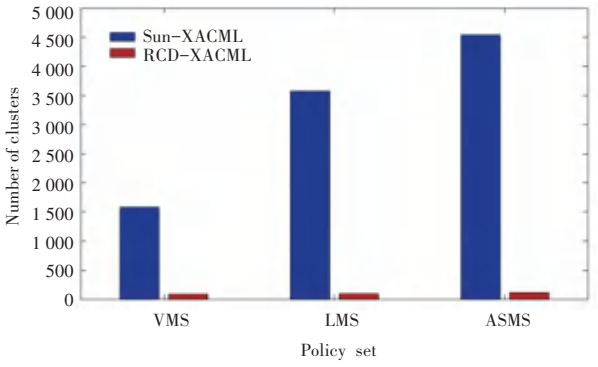


图 6 评估一个请求比较的规则

Fig. 6 Number of rules for evaluating a request

5 结束语

本文通过 XACML 策略评估技术的现状,认识到该技术在处理速度和效率上面临着严峻的挑战。为缓解目前的瓶颈,提出分布式 XACML 数据流模型,通过改善 PAP、PDP 两个模块优化 XACML 模型。首先提出规则原子化方法,根据规则状态及其相关性等概念来消除规则冗余状态。在规则聚类方面,基于规则间的相似性,使用 Chameleon 算法将策略集中的规则划分为多个簇,并结合分布式 PDP 体系结构,将规则簇分别加载到各个 PDP 中,使得访问请求与其最可能适用的规则进行匹配,减少规则和请求之间的比较次数,且多个 PDP 可同时独立工作,极大地提高了评估效率。实验结果表明,与目前广泛使用的 Sun-XACML 相比,RCD-XACML 评估性能的改进尤为显著。

参考文献

[1] DENG Fan, YU Zhenhua, LIU Wenjing, et al. An efficient policy evaluation engine for XACML policy management[J]. Information Sciences, 2021, 547: 1105-1121.

[2] DENG Fan, YU Zhenhua, ZHANG Liyong, et al. Clustering and supervised response for XACML policy evaluation and management[J]. Knowledge - Based Systems, 2020, 205: 106312.

[3] AL-SHARHAN S, OMRAN E, LARI K. An integrated holistic model for an eHealth system: A national implementation approach and a new cloud-based security model[J]. International Journal of Information Management, 2019, 7(8): 121-130.

[4] RIAD K, CHENG J. Adaptive XACML access policies for heterogeneous distributed IoT environments[J]. Information Sciences, 2021, 548: 135-152.

[5] 夏桐. 属性访问控制策略的自动生成与高效评估研究[D]. 昆明:云南师范大学, 2024.

[6] 牛嘉民. 访问控制高效策略评估方法研究[D]. 西安: 西安电子科技大学, 2023.

[7] TURKMEN F, DEMCHENKO Y . On the use of SMT solving for XACML policy evaluation[C]// Proceedings of 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). Piscataway, NJ: IEEE, 2016: 539-544.

[8] 潘瑞杰, 王高才, 黄珩逸. 基于属性访问控制策略管理方法[J]. 计算机工程与设计, 2022, 43(3): 601-607.

[9] DENG Fan, ZHOU Boyu, ZHANG Jiawei, et al. Elimination of the redundancy related to combining algorithms to improve the PDP evaluation performance[J]. Mathematical Problems in Engineering, 2016, 2016: 7608408.

[10] 王雅哲, 冯登国, 张立武, 等. 基于多层次优化技术的 XACML 策略评估引擎[J]. 软件学报, 2011, 22(2): 323-338.

[11] 罗杨, 沈晴霓, 吴中海. 一种基于元模型的访问控制策略描述语言[J]. 软件学报, 2020, 31(2): 439-454.

[12] 刘涛. 支持 XACML 标准的跨链安全访问控制关键技术研究[D]. 北京:北京邮电大学, 2023.

[13] 刘敖迪, 杜学绘, 王娜, 等. 基于访问控制日志的访问控制策略生成方法[J]. 电子与信息学报, 2022, 44(1): 321-331.

[14] 郑楷, 田秀霞, 卢官宇, 等. 基于聚类和重排序的 XACML 策略评估优化方法[J]. 计算机仿真, 2022, 39(10): 519-525.

[15] 邵瑞雪, 田秀霞. 智能电网中基于 MQTT 协议的 ABAC 访问控制方案[J]. 计算机应用研究, 2022, 39(11): 3436-3443.

[16] 吴嘉余. 雾计算下基于风险的 XACML 访问控制模型研究及实现[D]. 南京:南京邮电大学, 2021.

[17] 张玉秀, 王勇, 周林. 基于 FSA 和重排序的访问控制策略评估方案[J]. 计算机工程与设计, 2023, 44(3): 685-691.

[18] MENG Yang, SHANG Ronghua, JIAO Licheng, et al. Feature selection based dual-graph sparse non-negative matrix factorization for local discriminative clustering[J]. Neurocomputing, 2018, 290: 87-99.

[19] 黄恒杰, 潘瑞杰, 王高才, 等. 基于二进制序列的属性访问控制策略检索方法[J]. 计算机应用研究, 2022, 39(10): 3141-3145.

[20] 张润南. 访问控制安全策略建模与特性评价方法研究[D]. 西安: 西安电子科技大学, 2022.

[21] MEJRI M, YAHYAOUI H, MOURAD A, et al. A rewriting system for the assessment of XACML policies relationship[J]. Computers & Security, 2020, 97:101957.

[22] 裴文贤. ABAC 安全策略的异常检测方法研究[D]. 西安: 西安电子科技大学, 2021.

[23] 管哲. 基于类别和环境的访问控制策略语言的设计与实现[D]. 海口: 海南大学, 2022.

[24] KARYPIS G, HAN E H, KUMAR V. Chameleon: Hierarchical clustering using dynamic modeling[J]. Computer, 1999, 32(8): 68-75.