计算机网络防御策略模型

夏春和 魏玉娣 李肖坚 何 巍

(北京航空航天大学 虚拟现实技术与系统国家重点实验室, 北京 100191)

摘 要: 目前计算机网络防御研究中缺乏高层且易于细化的策略建模方法,因此在分析 Or-BAC 模型(Organization Based Access Control model)的基础上,对网络防御控制行为进行抽象,建立计算机网络防御策略模型(CNDPM, Computer Network Defense Policy Model). 该模型对保护、检测和响应等策略进行统一建模,并引入角色、视图、活动自动分配的方法,以提高分配的效率,同时给出了策略到规则的推导规则,以细化为具体的防御规则. 还给出了策略的完备性、有效性和一致性的形式化描述及分析. 实例分析表明,该模型表示的计算机网络防御策略,能够有效地转化为防御规则,具有较好的实用性和扩展性.

关 键 词: 计算机网络防御(CND); 策略; PPDR 模型; 细化

中图分类号: TP 393.08

文献标识码: A **文 章 编 号**: 1001-5965(2008)08-0925-05

Computer network defense policy model

Xia Chunhe Wei Yudi Li Xiaojian He Wei

(State Key Laboratory of Virtual Reality Technology and System, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: Recent research on computer network defense is lack of a method which is able to model policy in high level and refine policy conveniently, hence computer network defense policy model (CNDPM) was presented to abstract network defense control behavior on the basis of organization based access control model (Or-BAC). The CNDPM provides a common method to model protection, detection and response policy, and introduces automatic assignment mechanism of role as well as view and activity to improve efficiency, also provides derivation principles to refine policy to concrete defense rule. Moreover, completeness, validity and consistency of policy are studied through formal analysis. The example shows that computer network defense policies modeled by CNDPM can be refined to defense rules conveniently and efficiently. The CNDPM model is characterized by good expansibility and practicability.

Key words: computer network defense (CND); policy; PPDR model; refinement

计算机网络防御策略是指为了实现特定的安全目标,计算机网络和信息系统根据一定条件选择防御措施的规则[1-2]. 计算机网络防御必须使用大量的措施来应对网络攻击. 在网络环境下,特别是在大规模的环境下,这些措施的人工配置具有复杂、易错、低效的缺点,然而策略的使用不仅可以实现措施高效、正确、自动化的部署,还可以实现系统的可伸缩性和灵活性.

由于策略的许多优点,策略已经成为网络防御的核心,所有的保护、检测、响应都是依据策略实施,这就是著名的 PPDR 模型[3]的思想. 该模型中,策略粒度过大,无法实现基于策略的防御. 针对这个先天不足,策略树模型[4]将防御策略表示成目标-措施的形式,但是没有给出具体实施机制,存在兼容性不够好、层次不够清晰的问题.

本文所给出的计算机网络防御策略模型

收稿日期: 2007-09-06

基金项目: 北京教育委员会共建项目建设计划基金资助项目(JD100060517); 国家 863 计划资助项目(2007AA01Z407)

作者简介: 夏春和(1965—),男,江苏海安人,教授,xch@buaa.edu.cn.

(CNDPM, Computer Network Defense Policy Model),通过扩展 Or-BAC 模型(Organization Based Access Control model),能够对保护、检测和响应策略进行统一建模,并能细化到具体规则,如防火墙的 ACL 列表,IDS 检测规则的配置等,供计算机网络防御使用.

1 相关工作

文献[5]提出了策略分层描述的思想,文献[6]给出了4层策略模型,随着层次的增加,商业细节减少,技术细节增多.但是这些模型没有严格区分层次之间的界限和映射关系.

由 PCIM[7] 扩展的策略模型或语言 NSPIM[8],由于在层次上与 ACL 列表很接近,因而不能充分支持策略的高效、自动化的部署.

在访问控制领域,文献[9-11]给出了基于 RBAC 的网络访问控制策略模型.由于 RBAC 只将主体抽象为角色,权限仍然是具体的.针对该问题,文献[12]提出了 Or-BAC 模型,该模型在 RBAC 的基础上,将客体和动作分别抽象为视图和活动,能在统一框架中同时处理不同组织在多种环境下的策略.目前,Or-BAC 模型提出了一些概念和框架,然而只在访问控制[13]和响应[14]领域有进一步探索.

与以上模型相比,基于 Or-BAC 的 CNDPM 不仅能对访问控制策略建模,而且能对检测策略、响应策略及其它的保护策略统一建模,同时能转化为具体规则,而且具有良好的可扩展性,可以方便地对其扩展以表示更多的策略.

2 CNDPM 模型

本节首先深入分析概念及其关系,之后给出策略的形式化描述及推导,最后给出 CNDPM 模型图.

2.1 基本概念

2.1.1 组 织

定义 1 组织(organization)是组成部分及部分间有序关系的有机体[15].例如,医院、学校、计算机网络中的内网和外网等都是组织.

下面详细阐述各组分及其关系.

2.1.2 主体、角色及其关系

定义 2 主体(subject)指施加操作(改变系统状态或引起信息流动)的主动实体[16].记为

$$\begin{aligned}\mathrm{SUBJECT}::=\{&e\mid e\in \mathrm{ENTITY},\\ &\mathrm{Operational}(e)\wedge \mathrm{Active}(e)\}\end{aligned}$$

其中,Operational(e)和 Active(e)分别表示实体 e 是操作的、主动的.

主体可分为用户和节点,用户主要有用户名和口令两种特征;节点的特征主要有节点名、IP、掩码.

定义 3 角色(role)是拥有一些共同特征的主体集,且与一组权限或职责联系在一起[17],特征(feature)的总和即特性(character).记为

$$\begin{aligned}\mathrm{ROLE}::=\{&s\mid \mathrm{Own}(s,\mathrm{ch})\wedge\\ &(\mathrm{Relate}(\mathrm{ch},\mathrm{obligation})\vee \mathrm{Relate}(\mathrm{ch},\mathrm{right})),\\ &\mathrm{ch}\in \mathrm{CHARACTER}, s\in \mathrm{SUBJECT},\\ &\mathrm{right}\in \mathrm{RIGHT}, \mathrm{obligation}\in \mathrm{OBLIGATION}\}\end{aligned}$$

其中,SUBJECT 指主体类型;RIGHT 指权限类型;OBLIGATION 指职责类型;Own(s,ch)表示 s 拥有 ch;Relate(ch,right)表示 ch 与 right 相关;Relate(ch,obligation)表示 ch 与 obligation 相关.

由于角色是组织的组分,角色 r 与组织 org 相关记为 Relate(r,org).

从定义可看出,角色与主体存在成员关系,角色分配给主体有下面两种方式,Or-BAC 模型只给出第1种方式.

1) 显式分配

通过枚举将角色分配给主体.组织 org 雇用主体 s 作为角色 r,记为

$$\mathrm{Employ}(\mathrm{org},s)\wedge \mathrm{As}(s,r)$$

其中,谓词 Employ(org,s)表示 org 雇佣 s;As(s,r)表示 s 作为 r.由于这两个谓词经常同时出现,将其合成记为 EmployAs(org,s,r).

这种分配方式需要手动枚举,因此本文引入角色的定义将角色间接、自动地分配给主体.

2) 隐式定义

指定角色 r 的特性为 ch,记为

$$\mathrm{Define}(r,\mathrm{ch})\text{且}\ r\in \mathrm{ROLE};\mathrm{ch}\in \mathrm{CHARACTER}$$

更进一步,有如下推论:

推理 1 任何组织定义的任一角色、雇用的任一主体,如果主体具有此角色含有的特性,则该主体可以作为组织的角色.记为 R_1:

$$\begin{aligned}&\mathrm{Employ}(\mathrm{org},s)\wedge \mathrm{Define}(r,\mathrm{ch})\wedge\\ &\mathrm{Own}(s,\mathrm{ch})\rightarrow \mathrm{EmployAs}(\mathrm{org},s,r)\end{aligned}$$

2.1.3 客体、视图及其关系

定义 4 客体(object)指包含或接收信息的被动实体[16].

客体可分为操作系统、服务、文件、节点等.

定义 5 视图(view)指满足一组共同特征的客体集,记为

$$\text{VIEW}::=\{o\mid \text{ch}\in\text{CHARACTER},\ \text{Own}(o,\text{ch}),o\in\text{OBJECT}\}$$

其中 Own(o,ch)表示 o 拥有 ch.

例如,保护级别为绝密的文件可看作视图.由于网络节点既可以是主体,又可以是客体,因此用 ToView(r)表示将角色 r 作为视图对待.

类似地,Relate(v,org)表示 v 与 org 相关.组织 org 使用客体 o 作为视图 v,记为 Use(org,o) ∧ As(o,v),合成为 UseAs(org,o,v).视图的定义则可使用谓词 Define(v,ch).

推理 2　视图自动分配的规则记为 R_2:

$$\text{Use}(\text{org},o)\wedge\text{Define}(v,\text{ch})\wedge\text{Own}(o,\text{ch})\rightarrow\text{UseAs}(\text{org},o,v)$$

2.1.4　*动作、活动及其关系*

定义 6　动作(action)是不可再分的状态变化.

动作分为文件访问和服务访问,其中文件访问有创建、读、写和删除 4 种;服务访问包括协议和消息,消息中包含报头、内容和数目.

定义 7　活动(activity)是拥有共同特性的动作集.记为

$$\text{ACTIVITY}::=\{\alpha\mid\text{Own}(\alpha,\text{ch}),\ \alpha\in\text{ACTION},\text{ch}\in\text{CHRACTER}\}$$

例如 ftp,smtp,web 访问等都使用 tcp 协议,则这些动作可形成名为 alltcp 的活动.

活动 a 与组织 org 相关记为 Relate(a,org).活动的定义记为 Define(a,ch).组织 org 考虑动作 α 作为活动 a,记为 Consider(org,α) ∧ As(α,a),合成为 ConsiderAs(org,α,a).

推理 3　活动自动分配的规则记为 R_3:

$$\text{Consider}(\text{org},\alpha)\wedge\text{Define}(a,\text{ch})\wedge\text{Own}(\alpha,\text{ch})\rightarrow\text{ConsiderAs}(\text{org},\alpha,a)$$

2.1.5　*上 下 文*

定义 8　上下文(context)是组织中的主体给客体施加动作的具体环境.包括:时间、随时间而动态改变的特征、主客体及动作的关系.

组织 org 中,主体 s,客体 o 和动作 α 符合上下文 c 的要求,记为 Satisfy(org,s,α,o,c).

该谓词的真值可以根据 c 的表达式和 s,α,o 的特征自动判断.

2.1.6　*措　施*

定义 9　措施(measure)是保护、检测和响应未授权活动的动作,记为

$$M::=\{\alpha\mid\text{Protect}(\alpha)\vee\text{Detect}(\alpha)\vee\text{Respond}(\alpha),\alpha\in\text{Action}\}$$

其中 Protect(α),Detect(α),Respond(α)分别表示 α 是保护动作、检测动作、响应动作.

2.2　策　略

策略(policy)可以形式化描述成六元组:

$$\text{POLICY}::=<\text{org},r,a,v,c,m>$$

其中,org∈ORG;r∈ROLE;a∈ACTIVITY;v∈VIEW;c∈CONTEXT;m∈MEASURE.

策略中各实体存在偏函数的关系:

$$F:\text{ORG}\times R\times A\times V\times C\rightarrow M$$

2.3　具体规则

具体规则(rule)指主体对客体允许、不允许的动作或对不良行为的响应的声明.规则形式化描述为五元组:RULE::=<org,s,α,o,m>.表示组织 org 中主体 s 对客体 o 的动作 α 应用措施 m.

2.4　策略到具体规则的推导

给定策略,则雇用为角色 r 的主体 s,使用为视图 v 的客体 o,考虑为活动 a 的动作 α,如果处于上下文 c 中,则应用策略中的措施 m.记为 RD:

$$\forall p(p=<\text{org},r,a,v,c,m>\wedge\text{EmployAs}(\text{org},s,r)\wedge\text{UseAs}(\text{org},o,v)\wedge\text{ConsiderAs}(\text{org},\alpha,a)\wedge\text{Satisfy}(\text{org},s,\alpha,o,c))\rightarrow\text{rule}=<\text{org},s,\alpha,o,m>$$

2.5　模 型 图

综上所述,CNDPM 包含 9 种实体和 10 种关系,如图 1 所示.可以看出,策略与规则的实体通过谓词建立起关联,从而实现策略的推导.

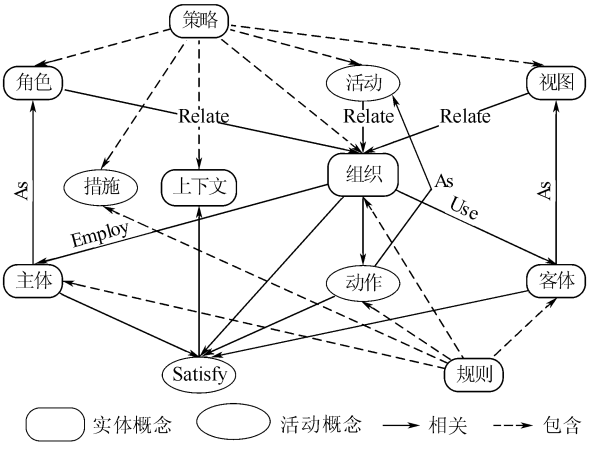


图 1　CNDPM 的 ER 图

3　策略的性质

策略的性质包括完备性、有效性和一致性[4].

3.1　完备性的构造

定义 10　如果任何组织,任何一个事件 INCIDENT 都至少对应一条策略,则称策略集是完备的.该性质的 Z 语言架构定义如图 2 所示.

为实现完备性,对于某些已知攻击,可以再定义一条缺省策略来构造.而未知攻击只能转化已知攻击时,才能添加检测和响应策略.

Completeness

I:INCIDENT
p:PolicySet

$\forall \text{org} \forall I \mid \text{org} \in \text{ORG} \wedge I \in \text{INCIDENT} \cdot$
$\exists p(p=<\text{org}, r, a, v, I, m>)$

图2 策略集完备性的定义

3.2 有效性的验证

定义11 任何策略的预期风险都在一个可接受的范围内,则策略集是有效的,如图3所示.

validity

p:PolicySet
$\text{AccessRisk}:\text{ORG}\times R\times A\times V\times C\times M\rightarrow N_1$
$\text{limit}:N_1$

$\forall p \mid p=<\text{org}, r, a, v, c, m> \cdot \text{AccessRisk}(p)\leqslant \text{limit}$

图3 策略集有效性的定义

3.3 一致性的检测

当两条策略的组织和上下文相同,而且角色、视图和活动都有交集,却采取了相冲突的措施,如禁止和许可,则会出现策略集不一致的现象.

定义12 p_1, p_2是策略集 P 中的任意两条策略,如果它们有相同的组织和上下文,且角色、视图和活动都有重叠,$p_1.m$ 和 $p_2.m$ 满足 CR 关系,$\text{CR}(m_1, m_2)$表示 m_1 和 m_2 不会冲突,则策略集P是一致的,记为 $C(P)$.如图4所示.

Consistency

p:PolicySet
$\text{CR}:M\times N\rightarrow\{0,1\}$

$\forall p_1, p_2\}(p_1.\text{org}=p_2.\text{org}\wedge p_1.c=p_2.c\wedge$
$p_1.r\cap p_2.r\neq\varnothing\wedge p_1.v\cap p_2.v\neq\varnothing\wedge$
$p_1.a\cap p_2.a\neq\varnothing)\cdot \text{CR}(p_1.m, p_2.m)$

图4 策略集一致性的定义

下面给出策略集一致性的判定命题.

命题1

$\forall p_1, p_2\in P\wedge(p_1.\text{org}\neq p_2.\text{org}\vee p_1.c\neq p_2.c\vee p_1.r\cap p_2.r=\varnothing\vee p_1.v\cap p_2.v=\varnothing\vee p_1.a\cap p_2.a=\varnothing)\Rightarrow C(P)$

证明 根据前提,任意两条策略的前件不同,则只能得到唯一的措施,策略集一致. **证毕**

由该定理可以得到关于其逆否命题的推论.

推论 策略集不一致,则组织和上下文相同,且角色、视图和活动有交集.

命题2

$\forall p_1, p_2\in P\wedge(p_1.\text{org}=p_2.\text{org}\wedge p_1.c=p_2.c\wedge p_1.r\cap p_2.r\neq\varnothing\wedge p_2.v\cap p_2.v\neq\varnothing\wedge p_1.a\cap p_2.a\neq\varnothing)$

则 $C(P)\Leftrightarrow\text{CR}(p_1.m, p_2.m)$.

证明

1) 充分性

如果 $\text{CR}(p_1.m, p_2.m)$,则这两条措施是兼容的,根据定义,p_1, p_2 不会发生冲突,策略集是一致的.

2) 必要性

偏函数 $F:\text{ORG}\times R\times V\times A\times C\rightarrow M$ 中,两条策略可能同时触发.如果$\neg$ $\text{CR}(p_1.m, p_2.m)$,则措施相互冲突,导致 p_1, p_2 不一致,从而得出策略集不一致的矛盾结论.因此 $\text{CR}(p_1.m, p_2.m)$是一个必要条件. **证毕**

根据命题1和命题2,可判断策略集是否具有一致性.发生不一致时,可用最先匹配优先法来解决.

4 实例分析

下面以具体的防御策略为例,说明该建模方法.

4.1 网络拓扑

如图5所示,虚线将网络 H 划分为4个区域,对应于防火墙的4个接口,每个接口有输入和输出2个方向:左边是外网;上方与网络IDS相连;下方的DMZ区包含2个外部可访问的服务器,其中dnsserver提供DNS服务,mulitiserver提供HTTP,FTP,E-mail服务;右边是内网.

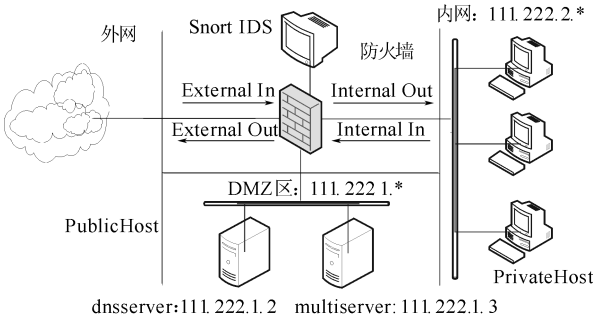


图5 网络拓扑实例

4.2 策略的建模

把网络 H 看成一个组织,H 的角色、视图和活动分别在表1、表2和表3给出描述,上下文有default和ftpoverflow_attack(ftp溢出攻击)两种类型.表4中给出了组织 H 中的策略.表5给出了策略所生成的配置在防火墙上的ACL列表.

表1 角色的描述

角色名	定义或分配
PublicHost	分配给外网中的任何主机

PrivateHost	ip=111.222.2.0∧mask=24

表 2　视图的描述

视图名	分　配
MultiServer	分配给 multiserver
DnsServer	分配给 dnsserver

表 3　活动的描述

活动名	定　义
alltcp	protocol=tcp
ftp	protocol=tcp∧dport=21
multitcp	protocol=tcp∧(dport=110∨dport=25∨dport=80∨dport=21)
dns	protocol=udp∧dport=53

表 4　组织 *H* 的防御策略

编号	防御策略
1	<*H*,PublicHost,dns,DnsServer,default,Permit>
2	<*H*,PublicHost,multitcp,MultiServer,default,Permit>
3	<*H*,PrivateHost,dns,DnsServer,default,Permit>
4	<*H*,PrivateHost,multicp,MultiServer,default,Permit>
5	<*H*, PrivateHost, alltcp, ToView(PublicHost), default, Permit>
6	<*H*,PublicHost,ftp,multiserver,default,ftpoverflow>
7	<*H*,PublicHost,ftp,MultiServer,ftpoverflow_attack,Prohibit>

4.3　具体规则的产生

先从策略 1 开始,表示允许角色 PublicHost 访问视图 DnsServer 的 dns 服务.

1) 角色 PublicHost(见表 1)分配给外网的每一台主机,对应于 ACL 列表中的源地址,表示为 any.

2) 由 dns 的定义(见表 3),及 R_3 规则可知,协议是 udp、目的端口为 53 的访问分配给该活动,动作对应于 ACL 列表中的协议和目的端口,分别是 udp 和 53. 由于 dns 服务的源端口是 53,所以 acl 中源端口也设为 53.

3) 视图 DnsServer(见表 2)只分配给 dnsserver 节点,对应于 ACL 列表的目的 IP,取值为 host 111.222.1.2

根据 RD,可得到 ACL1,而且配置 External In 接口.

类似于策略 1,策略 3 表示允许 PrivateHost 访问 DnsServer 的 dns 服务. 可推导出 ACL6. 策略 2、策略 4 和策略 5 也分别得到 ACL2～ACL5,ACL7～ACL10,ACL11.

策略 6、策略 7 表示对 ftp 溢出攻击的检测和响应策略. 由实际的攻击来触发.

表 5　防火墙上的 ACL 配置

编号	接口	Cisco ACL 列表
ACL1	External In	access-list 100 permit udp any eq 53 host 111.222.1.2 eq 53
ACL 2	External In	access-list 100 permit tcp any eq 80 host 111.222.1.3 eq 80
ACL 3	External In	access-list 100 permit tcp any eq 21 host 111.222.1.3 eq 21
ACL 4	External In	access-list 100 permit tcp any eq 25 host 111.222.1.3 eq 25
ACL 5	External In	access-list 100 permit tcp any eq 110 host 111.222.1.3 eq 110
ACL 6	Internal In	access-list 101 permit udp 111.222.2.0 0.0.0.255 eq 53 host 111.222.1.2 eq53
ACL 7	Internal In	access-list 101 permit tcp 111.222.2.0 0.0.0.255 eq 80 host 111.222.1.3 eq 80
ACL 8	Internal In	access-list 101 permit tcp 111.222.2.0 0.0.0.255 eq 21 host 111.222.1.3 eq 21
ACL 9	Internal In	access-list 101 permit tcp 111.222.2.0 0.0.0.255 eq 25 host 111.222.1.3 eq 25
ACL 10	Internal In	access-list 101 permit tcp 111.222.2.0 0.0.0.255 eq 110 host 111.222.1.3 eq 110
ACL 11	External Out	access-list 102 permit tcp 111.222.2.0 0.0.0.255 any any any

5　结　论

本文基于 Or-BAC 模型,详细分析了各种实体的概念及其关系,引入了自动分配的方法提高分配的效率,建立了 CNDPM 模型. 以形式化方法描述并论证了该模型下策略的完备性、正确性和一致性,并给出了具体网络拓扑下的实例分析. 理论分析和实例揭示了策略的建模方法及到防御规则的转化机制,为计算机网络防御策略描述语言的设计和防御规则的自动生成提供了重要的理论基础.

参考文献 (References)

[1] Sloman M S. Policy driven management for distributed systems[J]. Journal of Network and Systems Management, 1994, 2(4):333－360

[2] Stern D F. On the buzzword "security policy"[C]//Security and Privacy. Los Alamitors: IEEE CS Press, 1991: 219－230

[3] 夏春和. 基于入侵诱骗的防御体系及关键技术研究[D]. 北京:北京航空航天大学计算机学院, 2003
Xia Chunhe. Research on intrusion-deception-based network defense architecture and key technology[D]. Beijing: School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, 2003(in Chinese)

[4] Zhang Feng, Qin Zhiguang, Zhou Shijie. Policy-tree based proactive defense model for network security[C]//Grid and Cooperative Computing. Berlin: Springer, 2004, 3252: 437－439

[5] Moffett J D, Sloman M S. Policy hierarchies for distributed systems management[J]. IEEE JSAC Special Issue on Network Management, 1993, 11(9): 1404－1414

(下转第 943 页)