计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
48(2): 306-316, 2011

# 基于非合作动态博弈的网络安全主动防御技术研究

林旺群 王 慧 刘家红 邓 镭 李爱平 吴泉源 贾 焰
(国防科学技术大学计算机学院 长沙 410073)
(linwangqun@nudt.edu.cn)

# Research on Active Defense Technology in Network Security Based on Non-Cooperative Dynamic Game Theory

Lin Wangqun, Wang Hui, Liu Jiahong, Deng Lei, Li Aiping, Wu Quanyuan, and Jia Yan
(*College of Computer, National University of Defense Technology, Changsha* 410073)

**Abstract** Game theory, an important part of artificial intelligent technique, has been applied on network defense very well. Static model has been used widely in most of the previous studies. However, some work shows such model cannot follow the evolving of the strategies of attackers. In this paper, an active defense model based on dynamic game theory of non-cooperative and complete information has been given, that is, the attack-defense game tree has been generated by adding some virtual nodes on the original attack-defense graph. Based on the attack-defense game tree, the best defense strategies are achieved under current network environment through resolving the Nash equilibrium in different situations. Besides, for the scenarios with complete information and incomplete information, two algorithms have been proposed respectively. The analysis and experimental results show that the complexity of the algorithms can be guaranteed not worse than other similar works. Moreover, not only for scenario with complete information, but also in incomplete cases, the sensible results can be found. With the comparison of mixed strategy Nash equilibrium generated by static game and described in a probabilistic form, results given by the sub-game perfect Nash equilibrium are more easily to be understood and operated. Network research based on game theory should have a good application in the future network security product.

**Key words** network security; active defense; dynamic game theory; attack-defense game tree; Nash equilibrium

**摘 要** 目前基于博弈的网络安全主动防御技术大多采用静态博弈方式.针对这种静态方式无法应对攻击者攻击意图和攻击策略动态变化的不足,基于非合作、非零和动态博弈理论提出了完全信息动态博弈主动防御模型.通过"虚拟节点"将网络攻防图转化为攻防博弈树,并给出了分别适应于完全信息和非完全信息两种场景的攻防博弈算法.理论分析和实验表明相关算法在复杂度不高于同类算法的前提下:1)不仅适应于完全信息博弈场景,而且在非完全信息的特殊场景下仍能够得到合理的解;2)与采用静态博弈给出的以概率形式描述的混合策略Nash均衡解相比,给出的从子博弈精炼Nash均衡中抽出的解具有更好的可理解性和可操作性.

**关键词** 网络安全;主动防御;动态博弈;攻防博弈树;纳什均衡

**中图法分类号** TP393.08

**收稿日期**:2009-10-08;**修回日期**:2010-03-11

**基金项目**:国家"八六三"高技术研究发展计划基金项目(2007AA01Z474,2006AA01Z451,2007AA010502)

现有的入侵检测、防火墙技术大多采取安全策略配置的方式来保护网络的安全性.然而,这些静态被动防御技术在面对网络中未知攻击、瞬时攻击时,无法有效地完成动态的安全保障.理想的防御系统应该对所有的弱点或攻击行为都作出防护,但是这种"不惜一切代价"的防御显然是不合理的,因此必须考虑被保护系统的"适度安全"的概念[1].基于博弈理论的实时防御模型和技术的目标是通过态势感知、指标体系、风险评估等手段对当前网络安全态势进行判断和预测,并依据判断结果建立网络主动防御的主动安全防护体系[2].与传统的入侵检测等被动防御技术相比,主动防御技术能够帮助用户预先识别网络系统脆弱性以及所面临的潜在的安全威胁,以尽可能小的代价获取最大的网络安全保障.目前基于博弈理论的主动防御大多采用静态博弈模型.由于这类模型只需要根据攻防图进行收益量化形成收益矩阵即可计算 Nash 均衡解,因此具有计算简单、攻防双方只需一次博弈等优点.然而当攻击者在攻击过程中由于攻击兴趣或者攻击偏好发生变化而改变攻击手段和攻击意图时,基于静态博弈的主动防御模型就显得"无能为力".基于非合作动态博弈的主动防御模型能够根据网络攻防过程中攻击者攻击意图和攻击策略的改变,实时提供最佳防御策略,因而具有更好的适用性和灵活性.本文面向特定领域的网络安全,提出了基于非合作、动态博弈理论主动防御相关模型和算法.与已有的工作相比,在相关算法的算法复杂度不高于同类算法的前提下,本文提出的模型和算法具有更强的适用性,所提供的最佳攻防策略解具有更好的可理解性和可操作性,同时策略预测准确性得到进一步的提高.

## 1 相关工作

Agah 等人[3]将无线传感器网络中的 DoS 攻击和防御建模成入侵检测系统与节点间的重复博弈过程,通过设计新的通信协议识别恶意节点,以便更好地阻止无线传感器网络中 DoS 攻击.Hadi 等人[4]在网络数据包采样率必须满足实时性要求的前提下,通过建立零和、非合作博弈模型,分析了单节点攻击和多节点相互协作攻击的入侵检测两种场景.Liu 等人[5]基于贝叶斯模型理论分析攻防双方形成的 Nash 均衡,详细讨论了现实情况的动态贝叶斯模型.Baras 等人[6]基于不完全信息的重复博弈理论,将合法用户与非法用户的交互行为看成一种零和博弈,通过不断博弈过程试图找出具有非法行为的邻居节点,并将非法用户的破坏性控制在最小范围.Kotenko[7]模拟了分布式网络环境下多 Agent 的攻击方与多 Agent 协同作战的防御方之间的博弈,在模拟环境下验证了防御方之间的相互合作能够更有效的抵御攻击.Chen 等人[8]分析在静态贝叶斯博弈和动态贝叶斯博弈环境中攻防双方的 Nash 均衡,提出一种基于贝叶斯博弈模型的混合防御框架.在该框架中轻量级监测系统用来评估对手行为,重量级监测系统被用来作为最后的防御手段,最后通过实验验证了该框架能够提高无线传感器网络系统安全的同时还能节省节点本身能耗.Lye 等人[9]将攻防双方博弈建模为一个五元组,基于 Markov 链技术预测最可能的攻击行为和最佳防御策略,不足之处在于所提出的求解 Nash 均衡解算法的状态节点数量随可选攻防策略数量增加而成指数增长.

国内 Ma 等人[10]为了节省无线传感器网络节点能量,采用非合作博弈框架来决定每个节点簇的头节点以多大的概率来启动 IDS 服务,通过实验验证了提出的方法在节能和入侵检测率两者之间找到了一个平衡.姜伟等人[1]将网络攻击方和防御方相互博弈的过程看成一种两角色、非合作零和博弈,建立了一个攻防博弈模型(attack-defense game,ADG),不足之处在于所提模型无法适应攻击策略动态变化的场景.石进等人[11]在充分考虑博弈双方收益的基础上,提出了一种基于攻击图的入侵响应模型,通过两阶段决策方法给出防御方的最佳响应策略,该方法没有考虑攻防博弈全局信息,因而所给的解只能达到局部最优.另外,对于策略选取只有"选取"或"不选取"两种方案的主动防御来说,按照文献[1]和文献[9]的方法所得到的以概率形式给出的最佳攻防策略解具有较低的可理解性和可操作性.

## 2 定义描述

**定义 1.** 完全信息动态博弈主动防御模型(active defense model based on dynamic game theory of complete information, ADDG)用四元组$(N,H,P,U)$来描述,其中:

1) $N\triangleq\{p_1,\cdots,p_n\}$表示参与博弈的局中人集合.在网络攻防博弈中局中人是策略选择的主体和策略制定者,在大部分网络攻防博弈中可以将博弈双方看成是攻击者 $P_a$ 和防御者 $P_d$ 的二人博弈.

2) $H\triangleq\{h/h\triangleq(a^k)_{k=1}^{K}\}$, $a^k$ 是某一参与网络攻防博弈的局中人采取的策略方案,这个局中人根据 $P$ 由 $(a^j)_{j=1}^{k-1}$ 来定,($P$ 的定义见 3)),称为历史集.

3) $P:H\backslash Z\rightarrow N$ 是局中人函数,其中 $Z$ 是 $H$ 中这样的 $h$ 形成的子集合,$h=(a^k)_{k=1}^{+\infty}$(即 $k=+\infty$)或 $h=(a^k)_{k=1}^{K}$ 而不存在 $a^{K+1}$,使得 $(a^k)_{k=1}^{K+1}\in H$.

4) $U:Z\rightarrow R$ 是局中人效用函数,$i=1,\cdots,n$. 效用函数决定了局中人在采取相关策略后所得的收益大小. 在网络攻防博弈中,攻防双方的收益通常用收益向量表示.

历史集实际上就是参与网络攻防博弈的局中人已选择的策略集合. 根据历史集信息由局中人函数 $P$ 确定由哪个局中人进行策略决策. 对于网络系统的攻击方和防御方来说,策略的选取遵循一定的先后顺序,如攻击者在攻击策略集 $A_a$ 中选择一个或多个攻击策略集$\{a_a^i,a_a^j,a_a^k\cdots\}$(其中 $0<i,j,k<n$, $n$ 为攻击策略总数),防御者在防御策略集 $A_d$ 中选取防御策略集$\{a_d^{i'},a_d^{j'},a_d^{k'}\cdots\}$($0<i',j',k'<m$, $m$ 为防御策略总数). 攻防双方所采取的策略手段对于另外一方可以通过扫描、探测等技术手段获取,也就是说攻防双方是完全信息的. 在假设攻防博弈双方都为理性者的前提下,这种网络攻防双方相互博弈的过程可以建模为一种完全信息的多阶段动态博弈过程.

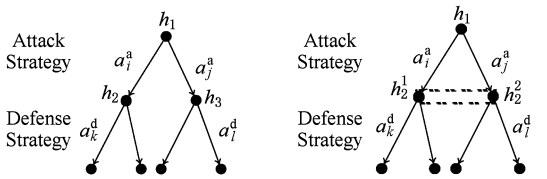
**定义 2.** 攻防博弈树 $T$ 具有一般树的结构,用三元组$(V,E,U)$表示. 其中 $V$ 表示所有节点集合,不同节点代表网络攻防中的物理节点的不同状态;$E$ 表示所有有向边集合,有向边代表攻击或防御策略;$U$ 表示所有收益向量集合,代表在不同攻防策略下攻防双方所取得的收益.

**定义 3.** 攻击方和防御方的一个信息集 $h$ 是指满足以下条件的状态节点的集合:

1) 在此信息集 $h$ 中的每一个状态节点 $s_i$ 都轮到局中人行动;

2) 当博弈进行到信息集 $h$ 中的一个状态节点时,局中人并不知道到达了信息集中哪一个节点.

在图 1(a)所示攻防博弈树(关于攻防博弈树的介绍见第 3 节)中攻击方有一个信息集 $h_1$,并且这个信息集包含一个决策状态节点. 防御方有两个信息集分别为 $h_2$ 和 $h_3$,每个信息集也各包含一个决策状态节点. 在图 1(b)所示攻防博弈树中,攻击方包含一个决策状态节点的信息集 $h_1$,而防御方包含一个具有两个决策状态节点的信息集 $h_2$,这两个决策状态节点分别为 $h_2^1$ 和 $h_2^2$ 所在的节点. 从信息集的概念中可以得出,在完全信息的动态博弈中,每个信息集都是单节点信息集,即每一个节点都构成一个信息集.




(a) Attack-defense tree with single node information set (b) Attack-defense tree with multi-node information set

Fig. 1 Attack-defense trees.

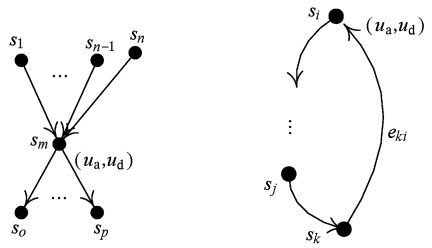
图 1 攻防博弈树

## 3 多阶段非合作动态博弈

网络攻防图 $G(V',E',U')$是一个有向带环图,其中 $V'$表示所有物理节点的所有状态集合(称为状态节点);$E'=\{E_a,E_d\}$代表网络攻防图 $G$ 的有向边集合,该集合中的每条边对应于网络攻击方或防御方所采取的攻防策略;$U'$表示网络攻防策略集所对应的收益向量集合. 网络攻防图是在网络攻击图的基础上加入防御策略后形成的. 关于网络攻击图的生成及其攻防策略收益量化参见文献[1,12-14],这里不作详细介绍.

### 3.1 攻防博弈树的生成

完全信息网络攻防动态博弈借助于攻防博弈树 $T(V,E,U)$进行描述和分析. 这是因为攻防博弈树能够描述攻击方和防御方的策略动态选择,也就是参与者在什么时候行动,此时参与者能获得什么样的信息及其收益,以及参与者又有哪些行动策略可供选择. 在完全信息的网络攻防动态博弈中攻防博弈树 $T$ 的表述还包括以下元素:1)参与者集合(攻击方 $p_a$ 和防御方 $p_d$);2)每个参与者在什么时候行动,即行动的先后顺序;3)每次轮到某一参与者行动时,可供他选择的策略集 $A_a$ 或 $A_d$;4)每次轮到某一参与者行动时他所了解到的信息;5)每个参与者在选择每一种行动策略所构成的策略组合时,各个参与者的所获得的收益. 由于网络攻防图 $G$ 是一个带环的有向图,为了将其应用于动态博弈场景,需要将网络攻防图 $G$ 转化为攻防博弈树 $T$. 攻防图 $G$ 中主要有两种类型的子图需要进行转换:①包含入度为 $n(n>1)$的节点的子图 $G_1(V_1,E_1,U_1)$,如图 2(a)所示;②包含环路的子图 $G_2(V_2,E_2,U_2)$,如图 2(b)所示:

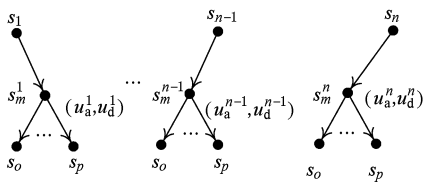



(a) The firs type subgraph (b) The second type subgraph

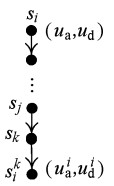
Fig. 2 The first and second type of subgraphs.

图 2 第 1 类子图和第 2 类子图

为了消除网络攻防图与攻防博弈树结构上的差异,通过引入"虚拟节点"将网络攻防图转换为攻防博弈树.具体方法如下:1)对于第 1 种类型的子图 $G_1$,将入度为 $n$ 的节点 $s_m$ 替换为包含 $n$ 个入度为 1 而出度保持与 $s_m$ 相同的虚拟节点 $s_m^1,\cdots,s_m^{n-1},s_m^n$;对于第 2 种类型的子图 $G_2$,将环路中由高级状态的节点 $s_k$ 指向低级状态节点 $s_i$ 的边 $e_{ki}$ 直接指向新引入的虚拟节点 $s_k^i$;2)对所有新加入的虚拟节点按照攻防策略量化模型进行收益量化处理.上述 1)中所说的高级状态节点向低级状态节点转换过程是指被攻击节点从正常状态 $s_0$ 到不同的非正常状态 $s_1,\cdots,s_{n-1},s_n$ 转换过程的逆过程.因此一般来讲,边 $e_{ki}$ 是防御者采取防御策略集使得被攻击节点状态向正常状态转变.图 3 分别表示两种不同类型的子图经过转换后形成的新的子图.




(a) The first type of subgraph after being transformed




(b) The second type of subgraph after being transformed

Fig. 3 Subgraphs after being transformed.

图 3 经过转换后的子图

**定理 1.** 由网络攻防图 $G(V',E',U')$ 通过引入虚拟节点形成的攻防博弈树 $T(V,E,U)$ 的节点规模 $|V|\leqslant(|V'|+|E'|)$,边的规模 $|E|\leqslant(|E'|/2)^2$.

证明. 在网络攻防图 $G$ 转换为攻防博弈树 $T$ 过程中,假设第 1 种类子图 $G_1(V_1,E_1,U_1)$ 新增加的节点数为 $n_1$,新增加的边数为 $e_1$;第 2 类子图 $G_2(V_2,E_2,U_2)$ 新增加的节点数为 $n_2$,新增加的边数为 $e_2$.

对于第 1 种类型子图 $G_1$,设其中多入度节点 $s_m$ 入度为 $d_{\text{in}}$,出度为 $d_{\text{out}}$,则有 $2\leqslant d_{\text{in}}\leqslant|E_1|-1$,$d_{\text{out}}=|E_1|-d_{\text{in}}$.引入虚拟节点后,新增加节点个数 $n_1=d_{\text{in}}-1\leqslant|E_1|-2$,新增加的边数 $e_1=n_1$.$d_{\text{out}}=(d_{\text{in}}-1).(|E_1|-d_{\text{in}})\leqslant(|E_1|-1)^2/4$.对于第 2 种类型子图 $G_2$,由引入虚拟节点的方法可知,转化 $G_2$ 将新增加节点 $n_2=1$,而新增加的边 $e_2=0$.假设攻防博弈图 $G$ 中包含第 1 种类子图 $G_1$ 为 $x$ 个(其中 $0<x\leqslant|E'|$),第 2 类子图 $G_2$ 为 $y$ 个(其中 $0<y\leqslant|E'|$),则求解攻防博弈树 $T$ 的节点规模和边的规模等价于求解下列规划解:

$$\max|V|=\max(|V'|+x.(|E_1|-2)+y),$$
$$\max|E|=\max(|E'|+x.(|E_1|-1)^2/4),$$
其中:$x.|E_1|+y.|E_2|\leqslant|E'|$. (1)

求解上述规划可知:

$$\max|V|=|V'|+|E'|-2|E'|/|E_1|,$$
$$\max|E|=|E'|+|E'|.(|E_1|-2+1/|E_1|)/4, \quad (2)$$

再由 $|E_1|\leqslant|E'|$ 可知 $\max|V|\leqslant|V'|+|E'|$,$\max|E|\leqslant(|E'|/2)^2$ 成立,命题得证. 证毕.

通过引入虚拟节点,成功地将网络攻防图 $G(V',E',U')$ 转换为动态博弈分析所需要的攻防博弈树 $T(V,E,U)$,并且博弈树 $T$ 节点数目不超过 $(|V'|+|E'|)$,边的数目不超过 $(|E'|/2)^2$(其中符号 $|V'|$ 和符号 $|E'|$ 分别表示网络攻防图 $G$ 中状态节点个数和有向边的条数).在实际情况中,由于攻防图 $G$ 所包含的上述两种类型子图一般较少,因此攻防博弈树 $T$ 的节点规模和边规模一般稍大于攻防图 $G$ 的规模.

### 3.2 多阶段动态博弈算法

在完全信息假设前提下,结合攻防博弈树,首先给出一种完全信息多阶段博弈逆向归纳算法.

**算法 1.** 完全信息多阶段动态博弈逆向归纳法.

输入:完全信息博弈的攻防博弈树 $T(V,E,U)$;

输出:最可能的攻击路径以及对应的最佳防御集 $S'$ 及其收益值 $u_{\text{all}}$.

① Initialize $G$;

② for ($k=H$; $k>1$; $k--$){

③ if $node_k.type==Defense\_node$ {

/ * 如果 $k$ 层节点类型为防御节点 * /

④ $u_{\text{temp}}=\max\limits_{u_{dj}}u(u_{ai},u_{dj})$;

⑤ $s\leftarrow s\bigcup node_{j,k}$ 的入弧;

⑥ if $node_{k-1}.type==node_k.type$ /* 若该节点所在的子博弈树上层节点也是防御节点 */

⑦ $node_{j,k}.father.u+=node_{j,k}.\boldsymbol{u}*\beta$;}

⑧ else{ /* 攻击节点 */

⑨ $u_{\text{temp}}=\max\limits_{u_{ai}} u(u_{ai},u_{dj})$;

⑩ $s\leftarrow s\cup node_{i,k}$的入弧;

⑪ if $node_{k-1}.type==node_k.type$; /* 父节点也是攻击节点 */

⑫ $node_{i,k}.father.u+=node_{i,k}.\boldsymbol{u}*\beta$;}}

⑬ $u_{\text{all}}+=u_{\text{temp}}$;

⑭ 按照从根节点到叶节点的顺序,从 $s$ 中抽取所有路径 $s'$

⑮ return($s'$,$u_{\text{all}}$);

算法开始时首先将攻防博弈树进行初始化处理.按照从最大深度叶节点到根节点的顺序分别逆序编号为 $H,H-1,\cdots,1$,同一层次节点从左至右按照从 1 到 $n$($n$ 为该层次节点最大数)的顺序进行编号.同时还需对各层节点的类型进行相应的标示.假设第 $k(1<k<H)$ 层节点的类型为攻击决策节点,那么具有相同高度的同一子树的所有节点(可能包括叶节点)都表示为攻击节点,即 $\forall i\in[0,n]$ $node_{i,k}.type=Defense\_node$.算法中 $S$ 用来存储生成的最可能的攻击路径和对应的最佳防御方法,在程序开始时,$S$ 应初始化为空.$u(u_{ai},u_{dj})$ 表示攻击者采取第 $i$ 个攻击策略,防御者采取第 $j$ 个防御策略所对应的收益向量.折扣因子 $\beta$ 表示为攻击者对将来收益值的重视程度.一般来讲,攻击者在采取攻击策略时,其攻击路径的深度与其获得的攻击效益成正比,但与被防御方检测出来的概率成反比,因而大部分攻击者在策略选取时更多关注于当前收益值大小.引入折扣因子 $\beta$ 可以把将来某个时刻所获得的收益值 $u_i$ 折算成现在的收益值 $u_i'$,如式(3)所示:

$$u_i'=u_i+\beta\times u_{i-1}+\beta^2\times u_{i-2}+\cdots+\beta^{i-j}\times u_j, \tag{3}$$

其中,$node_i.type=node_h.type\wedge node_{i-1}.type\neq node_j.type(i\leqslant h\leqslant j)$.

算法 1 根据完全信息攻防动态博弈局中人函数 $P:H\backslash Z\rightarrow N$ 求出本阶段对应局中人的收益值,然后根据上层节点类型将该收益值转化为未来收益.算法运行过程从攻防博弈树的叶节点出发,逆向推理至攻防博弈树的初始节点,中间的每一步都产生始于某个决策节点 $node_{m,n}$ 的子博弈 Nash 均衡,而且这个 Nash 均衡一定是该子博弈的所有子博弈(始于决策节点 $node_{m,n}$ 的后续节点的子博弈)的 Nash 均衡.由子博弈精炼 Nash 均衡的定义[15]可知,算法 1 采用的逆向归纳法所求得出的中间解 $S$ 就是攻防博弈树的子博弈精炼 Nash 均衡.由于最佳防御策略 $S'$ 是从 $S$ 中抽取的路径集合,因而是攻防博弈树 $T$ 的子博弈精炼 Nash 均衡解保证了 $S'$ 的稳定性.

算法 1 适应的两个前提条件:1)理性人假设,即攻击者和防御者都是完全理性的,因而在每次策略选取时攻防双方都是选择自己能够获得最大期望收益的策略;2)完全信息假设,即攻击者和防御者双方信息具有对称性.然而,当理性人假设被违背,如参与博弈的攻击方选择策略与算法预测方向发生偏离时,防御者应该从始于他的决策节点的子博弈继续使用算法 1 进行逆向归纳预测.另外,当完全信息假设得不到有效保证时,如在分布式网络环境下,由于攻击者手段的更新或者防御者弱点扫描工具的限制使得防御者可能无法检测或者无法实时检测到对方的攻击行为时,这就使得攻击者和防御者之间的信息产生非对称性.在这种情况下可以采用一种更为通用的非完全信息多阶段动态博弈逆向归纳算法,算法的描述如下:

**算法 2.** 非完全信息多阶段动态博弈逆向归纳法.

输入:攻防博弈树 $T(V,E,U)$;

输出:最可能的攻击路径以及对应的最佳防御集 $S'$ 及其收益值 $u_{\text{all}}$.

① Initialize $G$;

② for($k=H$; $k>1$; $k--$){

③ if $node_k.type==Defense\_node${ /* 如果 $k$ 层节点类型为防御节点 */

④ if $node_{j,k}$是单节点信息集{

⑤ $u_{\text{temp}}=\max\limits_{u_{dj}} u(u_{ai},u_{dj})$; /* 求出该层节点收益矩阵中防御者的最大值 $u_{dj}$ */

⑥ $s\leftarrow s\cup node_{j,k}$的入弧;

⑦ if $node_{k-1}.type==node_k.type$; /* 上层节点也是防御节点 */

⑧ $node_{j,k}.father.\boldsymbol{u}+=node_{j,k}.\boldsymbol{u}*\beta$;} /* 父节点收益值增加子节点收益值的 $\beta$ 倍 */

⑨ else { /* $node_{j,k}$非单节点信息 */

⑩ $StaticNashE(g)$;}}

⑪ else {/* 攻击节点 */

⑫ if $node_{i,k}$是单节点信息集{

⑬ $u_{\text{temp}}=\max_{u_{ai}} u(u_{ai},u_{dj})$;

⑭ $s\leftarrow s\cup node_{i,k}$的入弧;

⑮ if $node_{k-1}.type==node_k.type$; /* 父节点也是攻击节点 */

⑯ $node_{i,k}.father.\boldsymbol{u}+=node_{i,k}.\boldsymbol{u}*\beta$;}

⑰ else { /* $node_{i,k}$非单节点信息 */

⑱ $StaticNashE(g)$;}}}

⑲ $u_{\text{all}}+=u_{\text{temp}}$;

⑳ 按照从根节点到叶节点的顺序,从 $s$ 中抽取所有路径 $S'$

㉑ return($s'$,$u_{\text{all}}$);

$StaticNashE(g)$ /* 求解静态博弈 Nash 均衡 */

① $\boldsymbol{u}=(E_a(a^*),E_d(a^*))$;

② maximize $\boldsymbol{u}$;

③ subject to:

④ for all $a\in A_a\cup A_d$;

⑤ $E_a(a^*/a)\leqslant E_a(a^*)\cap E_d(a*/a)\leqslant E_d(a^*)$;

⑥ $s\leftarrow s\cup a^*$ 所在的边;

⑦ $g.root.\boldsymbol{u}+=\beta*\boldsymbol{v}$.

算法 2 是在完全信息条件得不到满足的情况下,在算法 1 的基础上发展起来的.攻防博弈双方的非完全信息假设使得在攻防博弈树中形成多节点信息集.算法 2 的基本思想是在逆向归纳求解过程中将多节点信息集所包含的节点及其公共父节点所形成的子博弈树看成一个静态博弈过程.通过线性规划求解静态博弈的混合策略 Nash 均衡,并将得到的收益向量 $\boldsymbol{v}$ 乘以折扣因子 $\beta$ 后加入到子博弈树的根节点的收益向量 $g.root.\boldsymbol{u}$ 中.另外由于静态博弈求解 Nash 均衡过程中所得到的策略集 $a^*$ 所对应的攻防路径也应加入到归纳解 $S$ 中,从而使得在最后形成的最佳攻防策略 $S'$ 中可能包含静态博弈策略 $a^*$ 所在的边.

### 3.3 算法复杂度分析

给定网络攻防图 $G$ 采用算法 1 进行逆向归纳求解最可能攻击路径和最佳防御策略,其算法复杂度分析可分为 3 部分进行:1)将网络攻防图 $G(V',E',U')$ 生成攻防博弈树 $T(V,E,U)$.该过程只需要对整个网络攻防图 $G$ 进行一次扫描,并找出其中的包含环路和入度大于 1 的节点的子图,通过引入"虚拟节点"技术将攻防图转换成攻防博弈树.在将给定的网络攻防图使用邻接表存储的情况下,扫描过程实际就是深度优先遍历邻接表过程,因此其算法复杂度为 $O(|V'|+|E'|)$.2)攻防博弈树初始化(对应算法 1 中①).攻防博弈树初始过程也就是树的遍历过程,因此其算法复杂度为 $O(|V|+|E|)$.3)逆向归纳求解(对应算法 1 中②~⑫).博弈树 $T$ 的高度 $H$ 就是逆向归纳求解的循环次数,而每次循环求最大值 $\max u$ 的复杂度为 $\max(|A_a|,|A_d|)$,因此该过程的算法复杂度为 $O(H\times\max(|A_a|,|A_d|))$.由攻防博弈树的生成过程可知:$|V'|\leqslant|V|$,$|E'|\leqslant|E|$ 成立.综合上面的分析可知:将给定的网络攻击图转化为攻防博弈树,并采用算法 1 求解的算法复杂度为 $O(|V|+|E|+H\times\max(|A_a|,|A_d|))$.

算法 2 在算法 1 的基础上求解多节点信息集的静态博弈 Nash 均衡.已经证明,当该静态博弈是零和博弈时,其算法复杂度是多项式时间的,但是当该静态博弈是非零和博弈时,作为一道世界性难题已经在 2006 年被证明[16]求其 Nash 均衡解是 PPAD-complete 问题.在实际应用中,为了提高算法效率,要求博弈树中非单节点信息集尽可能少.另外当找到合适的 Nash 均衡解后就停止运行,不必找到所有 Nash 均衡解.当然采取一种允许 Nash 均衡解误差的多项式方法算法[17-18]也是可取的.

### 3.4 主动防御模型及其算法比较

将本文所提出的模型及其算法与文献[1,9,11]提出的主动防御方法从对局势信息要求、博弈类型、所求解的可操作性、算法复杂度等方面进行分析比较,结果如表 1 所示:

**Table 1 Comparison with Correlated Works**

**表 1 相关工作比较**

| Related Algorithms | Environment Requirement | Game Style | Operation | Time Complexity(Zero sum/Non-zero sum) |
|---|---|---|---|---|
| Ref[9] | Complete Information | Static Game | General | $O(\lvert V\rvert^{\lvert E\rvert})$; PPAD-Complete |
| Ref[11] | Complete Information | Two Steps Dynamic Game | Best | $O(\max(\lvert A_a\rvert,\lvert A_d\rvert)$ |
| Ref[1] | Complete Information | Static Game | General | Polynomial Time; PPAD-Complete |
| Algorithm 1 | Complete Information | Multi-Step Dynamic Game | Best | $O(\lvert V\rvert+\lvert E\rvert+H\times\max(\lvert A_a\rvert,\lvert A_d\rvert))$ |
| Algorithm 2 | Complete Information | Multi-Step Dynamic Game | Better | Polynomial Time; PPAD-Complete |

其中算法复杂度分零和博弈及非零和博弈两种情形进行比较.所求解的可操作性是指相关模型和算法最终提供给用户的最佳攻防策略是否具有较强的现实意义,如在网络攻防博弈中,策略集的选取只有"选"和"不选"两种方案,那么以概率形式给出的策略选取方案将使得用户无所适从,不知如何选取对应的攻防策略集,因而具有较差的可操作性.

## 4 应用实例

为了进一步阐述本文所提出的主动防御模型及其相关算法的有效性,通过部署如图 4 所示的实验场景进行模拟实验.实验环境主要由一台安装 Windows XP SP2 操作系统的 Web 服务器、一台安装 Linux 操作系统的文件服务器和一台安装 Windows 2000 Server 操作系统的数据库服务器以及安装 ISO 操作系统的 Cisco 路由器组成.另外 Web 服务器上运行 Tomcat 程序,文件服务器上运行 FTP 服务端程序,数据库服务器运行 Oracle 10g 程序,Cisco 路由器安装防火墙软件包.

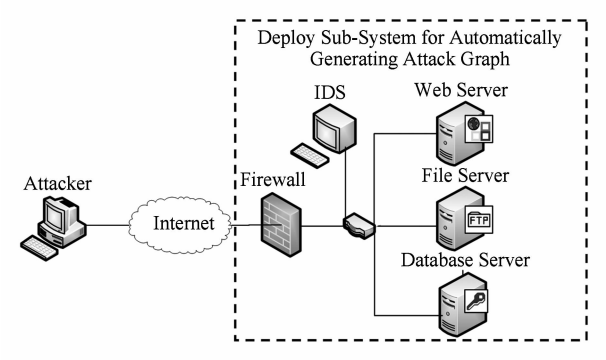



Fig. 4 The network environment of experiment.

图 4 实验环境网络拓扑结构

利用弱点扫描工具得到目标系统相关漏洞信息如表 2 所示.将攻击图自动生成子系统生成的攻击图在增加各个攻击阶段相应的防御措施后得到网络攻防图,然后按照 3.1 节提供的由攻防图生成攻防博弈树的方法得到如图 5 所示的攻防博弈树.在该博弈树中实线表示的有向边代表攻击方采取的攻击策略集,虚线表示的有向边代表防御方采取的防御策略集.另外有实线有向边引出的节点代表攻击节点;有虚线有向边引出的节点代表防御节点;既有实线有向边又有虚线有向边引出的节点代表混合节点,表示在该状态节点时既有可能防御方采取防御策略集,也有可能攻击方在该状态进行攻击.各个状态节点的含义如表 3 所示,攻防博弈树中攻击方和防御方可选的策略集如表 4 所示,攻防策略收益量化标准参考文献[1]提出的方法.在运行算法 1 时,为了满足完全信息假设,将博弈树中的所有混合节点都看成攻击节点,运行算法 2 时将包含混合节点的信息集所包含的所有状态节点及其父节点所形成的博弈看成一个静态博弈过程.

**Table 2 Description of Vulnerabilities in Experiment Environment**

**表 2 相关节点弱点信息描述**

| Node | Vulnerabilities | Vulnerability Description |
|---|---|---|
| Web Server (OS: Windows XP SP2) | ① LSASS LPC hole. (MS08-002)<br>② DNS client hole. (MS08-020)<br>③ TCP/IP hole. (MS08-004). | ① Advance privilege.<br>② Redirect Internet request.<br>③ Stop response and restart. |
| File Server (OS: Linux ) | ① Kernel file buffer overfolw hole. (Bugtraq ID: 26438)<br>② Hrtimers refuses serving hole. (Bugtraq ID: 26880) | ① Remote attacker may use this hole.<br>② Local attacker may make use of this hole and lead to refuse serving. |
| Database Server (OS:Windows 2000) | ① Listener GIOP's remotely refuse serving hole. (Bugtraq ID: 26103)<br>② MDSYS. SDO_CS's buffer overflow hole. (Bugtraq ID: 26243) | ① Listener core down and memory leak.<br>② Remote attacker may make use of this hole and control server. |
| Cisco Router (OS: ISO 12.3) | ① ACL hole. (Bugtraq ID: 26109)<br>② Cisco IOS hole. (Bugtraq ID:16303) | ① Lead to ACL being controlled.<br>② Lead to server refusing response and restarting automatically. |

运行算法 1 得到逆向归纳路径集合 $s=\{s^0\rightarrow s_w^1\rightarrow s_r^1, s^0\rightarrow s_w^7, s^0\rightarrow s_d^1, s^0\rightarrow s_d^3, s^0\rightarrow s_d^5, s_w^4\rightarrow s_w^5, s_w^3\rightarrow s_w^6, s_f^2\rightarrow s_f^4, s_f^2\rightarrow s_f^5, s^0\rightarrow s_f^6, s^0\rightarrow s_d^4\rightarrow s_d^2\rightarrow s_w^4\rightarrow s_w^3\rightarrow s_w^2, s^0\rightarrow s_f^3\rightarrow s_f^2\rightarrow s_f^1, s^0\rightarrow s_r^2\rightarrow s^0\}$,通过路径抽取过程得到的最佳攻防路径为 $s'=\{s^0\rightarrow s_r^2\rightarrow s^0\}$.由运行结果可以得出攻击方最可能采取策略集$\{a_a^2\}$对路由器采取拒绝服务攻击,而防御方的最佳防御策略集为$\{a_d^2\}$安装对应系统补丁.实验同时给出攻击方和防御方将

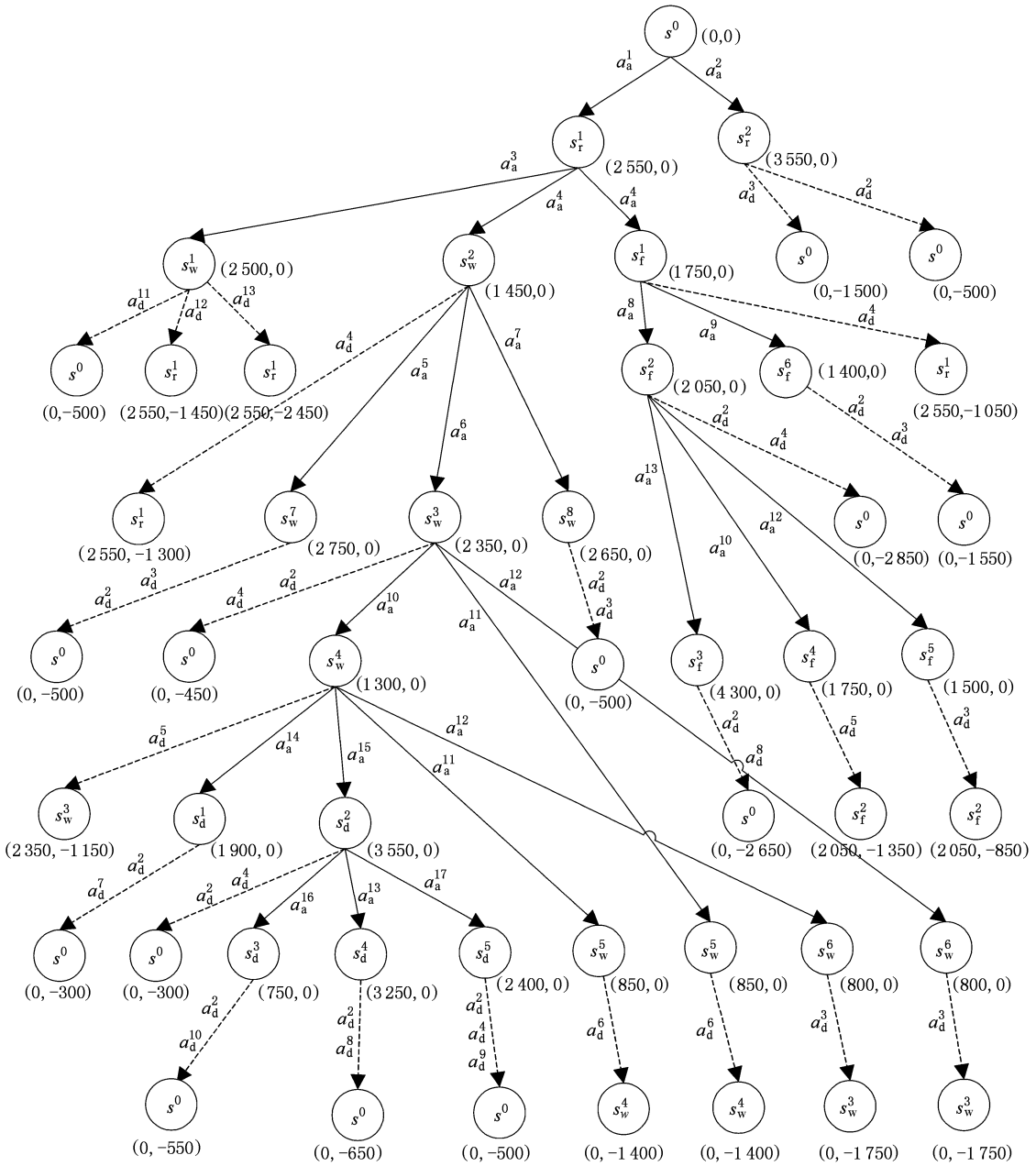



Fig. 5 Attack-defense game tree.

图 5 攻防博弈树

获得的收益向量为 $\boldsymbol{u}=(3\,550,-500)$. 当理性人假设被违背,即攻击者采取攻击策略集$\{a_a^1\}$突破防火墙后又选择攻击策略集$\{a_a^4\}$试图盗取并破解文件服务器账号文件时,实验运行结果为 $s'=\{s_f^1\rightarrow s_f^2\rightarrow s_f^3\rightarrow s^0\}$,$\boldsymbol{u}=(2\,332,-2\,650)$. 由实验结果可得出当攻击者绕过防火墙攻击文件服务器时,最可能利用文件服务器系统漏洞(Bugtraq ID: 26438)采用策略集$\{a_a^8\}$控制服务器,然后采用策略集$\{a_a^{13}\}$窃取或修改文件服务器数据. 这时防御方的最佳防御策略集为$\{a_d^2,a_d^8\}$,即安装文件服务器对应系统补丁,若文件服务器已经遭受到了数据篡改攻击,则还需要恢复文件数据.

为了测试算法 2 的效果,考虑位于博弈树第 6 层的混合状态节点 $s_d^2$,这时以 $s_d^2$ 为根节点的子博弈树的节点集$\{s_d^2,s_d^3,s_d^4,s_d^5\}$形成一个信息集. 为了与 $s_w^4$ 构成静态博弈,将节点 $s_d^2$ 和 $s_d^3$ 看成一个收益向量 $\boldsymbol{u}=(3\,700,0)$的状态节点 $s_d^{23}$;$s_d^2$ 和 $s_d^4$ 构成新的状态节点 $s_d^{24}$,收益向量 $\boldsymbol{u}=(4\,200,0)$;节点 $s_d^2$ 和 $s_d^5$ 构成新的状态节点 $s_d^{25}$,收益向量 $\boldsymbol{u}=(4\,030,0)$. 子博弈树构成的静态博弈过程如图 6 所示. 运行算法 2 得到的子博弈树静态博弈的最佳攻防路径 $s'=\{s_w^4\rightarrow$

$s_d^2 \rightarrow s_d^4 \rightarrow s^0\}$,得到的收益向量 $\boldsymbol{u}=\{4\,200,-650\}$. 从攻击路径可以看出,当攻击者到达状态节点 $s_w^4$ 在 Web 服务器安装木马后最可能采取策略集$\{a_a^{15}\}$利用数据库缓冲区漏洞(Bugtraq ID: 26243)控制数据库服务器,然后通过采取策略集$\{a_a^{13}\}$窃取或篡改数据库中的数据. 防御方的最佳防御策略集为$\{a_d^2, a_d^8\}$,即安装对应的数据库补丁,若数据库中的数据已经受到攻击则还需要实施数据恢复.

**Table 3 Description of State Nodes in Attack-Defense Game Tree**

**表 3 攻防博弈树中各状态节点描述**

| Node State | State Description | Node State | State Description |
|---|---|---|---|
| $s^0$ | Nodes_normal_working | $s_f^1$ | Fileserver_account_compromised |
| $s_r^0$ | Router_normal_working | $s_f^2$ | Fileserver_controlled |
| $s_r^1$ | Firewall_hacked | $s_f^3$ | Fileserver_data_stolen |
| $s_r^2$ | Router_refuse_serving | $s_f^4$ | Fileserver_install_malicious_progam |
| $s_w^1$ | Webserver_refuse_serving | $s_f^5$ | Fileserver_shutdown |
| $s_w^2$ | Webserver_account_compromised | $s_f^6$ | Fileserver_refuse_serving |
| $s_w^3$ | Webserver_controlled | $s_d^0$ | Database_server_normal_working |
| $s_w^4$ | Webserver_install_malicious_program | $s_d^1$ | Database_server_refuse_serving |
| $s_w^5$ | Website_defaced | $s_d^2$ | Database_server_controlled |
| $s_w^6$ | Webserver_shutdown | $s_d^3$ | Database_server_data_stolen |
| $s_w^7$ | Webserver_request_redirect | $s_d^4$ | Database_server_account_compromised |
| $s_w^8$ | Webserver_stop_serving | $s_d^5$ | Database_server_shutdown |
| $s_f^0$ | Fileserver_normal_working | | |

**Table 4 Symbols and the Corresponding Meanings of Strategies in Attack-Defense Game Tree**

**表 4 攻防博弈树中攻防策略集符号及其含义**

| Attack Strategy | Strategy Description | Defense Strategy | Strategy Description |
|---|---|---|---|
| $a_a^1$ | Make use of ISO system hole and control ACL | $a_d^1$ | Limit packets from related ports |
| $a_a^2$ | Send SGBP abnormal packet to server | $a_d^2$ | Install corresponding patches |
| $a_a^3$ | Run DDoS attack | $a_d^3$ | Restart server |
| $a_a^4$ | Steal account and crack it | $a_d^4$ | Delete suspicious account |
| $a_a^5$ | Request DNS sending special response | $a_d^5$ | Uninstall and delete Trojan program |
| $a_a^6$ | Send special LPC request to LSASS process | $a_d^6$ | Correct homepage |
| $a_a^7$ | Make use of TCP/IP hole | $a_d^7$ | Reinstall Lister program |
| $a_a^8$ | Make use of buffer hole 26438 | $a_d^8$ | Renew data |
| $a_a^9$ | Make use of system hole 26880 | $a_d^9$ | Limit access to MDSYS. SDO_CS |
| $a_a^{10}$ | Install Trojan | $a_d^{10}$ | Repair database |
| $a_a^{11}$ | Deface website, $a_a^{12}$: Shutdown server | $a_d^{11}$ | Redeploy firewall rule and filtrate malicious packet |
| $a_a^{12}$ | Shutdown server | $a_d^{12}$ | Limit SYN/ICMP packets |
| $a_a^{13}$ | Steal or tamper with data | $a_d^{13}$ | Add physical resource |
| $a_a^{14}$ | Send abnormal data to GIOP | $a_a^{15}$ | Make use of buffer hole 26243 |
| $a_a^{16}$ | Destroy database | $a_a^{17}$ | Get account and password maliciously |

为了将本文提出的方法与文献[1]提出的方法进行对比,对图 5 所示的攻防博弈树中第 2 层状态节点 $s_r^1$ 及其以最右子节点 $s_f^1$ 为根节点的子树形成的攻防博弈子树(实际上就是攻击者穿透防火墙后选择对文件服务器进行攻击所形成的攻防博弈树),分别运行文献[1]按照静态博弈模型提出的零和、静态博弈算法和算法 2 为了建立文献[1]提出的方法所需要的博弈矩阵,将攻击方对文件服务器进行攻击时可选攻击策略集及其防御方可选防御策略集按照文献[1]提出的方法进行量化处理形成攻防博弈

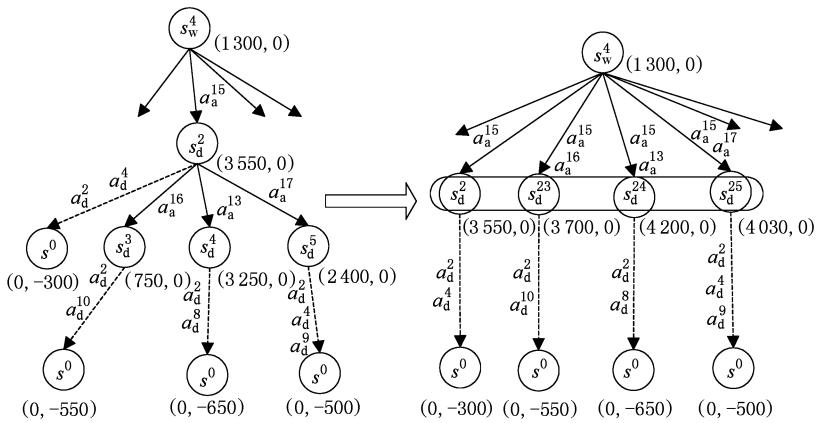



Fig. 6 Static game formed by mixed nodes.

图 6 混合节点形成的静态博弈

矩阵.在该矩阵中,可选攻击策略集为$\{\{a_a^4,a_a^8,a_a^{13}\},\{a_a^4,a_a^8,a_a^{10}\},\{a_a^4,a_a^8,a_a^{12}\},\{a_a^4,a_a^9\},\{a_a^4\}\}$,可选防御策略集为$\{\{a_d^2,a_d^8\},\{a_d^5\},\{a_d^3\},\{a_d^2,a_d^3\},\{a_d^4\}\}$.算法运行得到 Nash 均衡解:$P_a=(0,17/37,0,20/37,0)$;$P_d=(33/37,4/37,0,0,0)$,即攻击者的最佳攻击策略集是采取$\{a_a^4,a_a^9\}$进行拒绝服务攻击,防御者的最佳防御策略集为$\{a_d^2,a_d^8\}$,即安装相关补丁,并对文件服务器进行数据恢复.运行算法 2 得到的最佳攻防路径为 $s'=\{s_r^1\rightarrow s_f^1\rightarrow s_f^2\rightarrow s_f^3\rightarrow s^0\}$,收益向量 $\boldsymbol{u}=(2\,332,-2\,650)$.借助于图 6 所示的攻防博弈子树,从算法 2 给出的最佳攻防路径中可以得到防御者的最佳防御集为$\{a_d^2,a_d^8\}$,这与文献[1]提出的静态博弈方法得到的结果一致.然而按照文献[1]得出的攻击方最佳攻击手段为拒绝服务攻击,这与算法 2 给出的攻击方对文件服务器最可能的攻击为数据窃取和数据篡改(从攻击路径可以判断这一攻击意图)不一致.造成这种差异的原因在于当攻击方的攻击意图在攻击过程中发生改变时,由于采用静态博弈方式在攻防双方开始博弈前已形成了对局势的判断,因而无法应对这种攻击意图动态变化场景.如在图 5 所示的攻防博弈树中,攻击者在穿透防火墙后(到达状态节点 $s_r^1$),如若采取对 Web 服务器进行攻击(到达状态节点 $s_w^1$)理论上将获得更大预期收益,但是攻击者可能因为攻击偏好的改变等原因,选择了对文件服务器进行攻击(到达状态节点 $s_f^1$).由于算法 2 提出的动态博弈模型能够在实际攻击策略与预期攻击策略发生偏离时,在新的状态节点上重新进行动态博弈,因而在攻击意图发生改变时算法 2 仍能够给出更加合理的解.另外,算法 2 提出的方法考虑整个状态空间的博弈局势,以逆向推理的方式选取最优攻防策略,因而能够在全局范围内选取各个阶段的最佳攻防策略集.

## 5 结束语

针对传统的入侵检测、防火墙等被动防御技术无法应对日益突出的网络安全问题,本文提出了完全信息攻防动态博弈模型 ADDG,在网络攻防图的基础上引入“虚拟节点”技术,将网络攻防图转化为攻防博弈树.借助于攻防博弈树,采用非合作动态博弈理论解决网络安全主动防御的问题,并提出了相应的求解最佳攻防策略集的博弈算法.从理论分析和实验验证两方面将本文提出的方法与已有的工作进行对比,验证了本文工作的有效性.

由于网络攻防图的形成和攻防策略收益量化是本文的工作基础,攻防图生成的准确性和策略量化的合理性对基于动态博弈的主动防御所产生的最佳策略集有着至关重要的影响,因此下一步的工作将从以下两方面展开:1)高可靠网络攻防图的生成技术;2)网络资产风险评估技术.

## 参 考 文 献

[1] Jiang Wei, Fang Binxing, Tian Zhihong, et al. Evaluating network security and optimal active defense based on attack-defense game model [J]. Chinese Journal of Computers, 2009, 32(4): 817-827 (in Chinese)
(姜伟, 方滨兴, 田志宏, 等. 基于攻防博弈模型的网络安全测评和最优主动防御[J]. 计算机学报, 2009, 32(4): 817-827)

[2] Zhang Yongzheng, Fang Binxing, Chi Yue, et al. Risk propagation models for assessing network information systems [J]. Journal of Software, 2007, 18(1): 137-145 (in Chinese)

(张永铮, 方滨兴, 迟悦, 等. 用于评估网络信息系统的风险传播模型[J]. 软件学报, 2007, 18(1): 137-145)

[3] Afrand Agah, Das Sajal K. Preventing DoS attacks in wireless sensor networks: A repeated game theory approach [J]. International Journal of Network Security, 2007, 5(2): 145-153

[4] Hadi O, Mona M, Chadi A, et al. Game theoretic models for detecting network intrusions [J]. Computer Communications, 2008, 31(10): 1934-1944

[5] Liu Yu, Cristian C, Hong Man. A Bayesian game approach for intrusion detection in wireless ad hoc networks [C/OL] // Proc of Valuetools's 06. 2006: 11-13 [2010-03-08]. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.8169

[6] Theodorakopoulos G, Baras J S. Game theoretic modeling of malicious users in collaborative networks [J]. IEEE Journal on Selected Areas in Communications, 2008, 26(7): 1317-1327

[7] Kotenko I. Multi-agent modelling and simulation of cyber-attacks and cyber-defense for homeland security [J]. Information & Automation, 2007, 11(8): 614-619

[8] Chen Y, Boehm B, Sheppard L. Value driven security threat modeling based on attack path analysis [C] //Proc of the 40th Annual Hawaii Int Conf on System Sciences. Los Alamitos, CA: IEEE Computer Society, 2007

[9] Lye K, Wing J M. Game strategies in network security [J]. International Journal of Information Security, 2005, 4(1): 71-86

[10] Ma Yizhong, Cao Hui, Ma Jun. The intrusion detection method based on game theory in wireless sensor network [C/OL] //Proc of 1st IEEE Int Conf on Ubi-Media Computing. 2008: 326-331 [2009-10-01]. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4570911

[11] Shi Jin, Guo Shanqing, Lu Yin, et al. An intrusion response method based on attack graph [J]. Journal of Software, 2009, 19(10): 2746-2753 (in Chinese)
(石进, 郭山清, 陆音, 等. 一种基于攻击图的入侵响应方法[J]. 软件学报, 2009, 19(10): 2746-2753)

[12] Wang L, Islam T, Long T, et al. An attack graph-based probabilistic security metric [G] //LNCS 5094. Berlin: Springer, 2008: 283-296

[13] Wing J M. Attack graph generation and analysis [C/OL] // ACM Symp Information, Computer and Communications Security (ASIACCS'06). 2006: 14-14 [2009-10-01]. http://portal.acm.org/citation.cfm?id=1128822

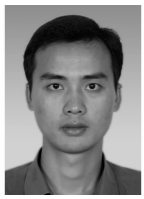
[14] Ou Xinming, Boyer Wayne F, McQueen Miles A. A scalable approach to attack graph generation [C] //Proc of the 13th ACM Conf on Computer and Communications Security (CCS'06). New York: ACM, 2006: 336-345

[15] Li Guangjiu. Basic Tutorial of Game Theory [M]. Beijing: Chemical Indusrty Press, 2005 (in Chinese)
(李光久. 博弈论基础教程[M]. 北京: 化学工业出版社, 2005)

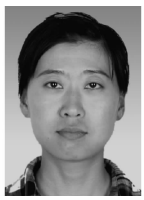
[16] Chen X, Deng X. Settling the complexity of two-player Nash equilibrium [C] //Proc of the 47the Annual IEEE Symp on Foundations of Computer Science(FOCS'06). Los Alamitos, CA: IEEE Computer Society, 2006: 261-272

[17] Kontogiannis S, Panagopoulou P, Spirakis P. Polynomial algorithms for approximating Nash equilibria of bimatrix games [J]. Theoretical Computer Science, 2009, 410(17): 1599-1606

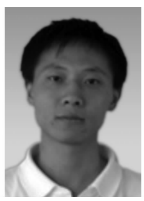
[18] Bosse H, Byrka J, Markakis E. New algorithms for approximate Nash equilibria in bimatrix games [G] //LNCS 4858. Berlin: Springer, 2007: 164-173



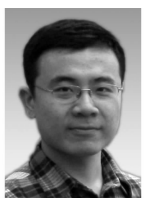
**Lin Wangqun**, born in 1983. PhD candidate. His main research interests include network security and data mining.



**Wang Hui**, born in 1981. PhD candidate. Her main research interests include network protocol design and stream media schedule.



**Liu Jiahong**, born in 1980. PhD candidate. His main research interests include event stream processing and service-oriented computing.



**Deng Lei**, born in 1984. PhD candidate. His main research interests include network security and game theory.

**Li Aiping**, born in 1974. Associate professor. His main research interests include network security, distributed computing and artificial intelligence.

**Wu Quayuan**, born in 1941. Professor and PhD supervisor in the National University of Defense Technology. His main research interests include distributed computing, artificial intelligence and network security.

**Jia Yan**, born in 1960. Professor and PhD supervisor of National University of Defense Technology. Her main research interests include distributed computing and network security.