第 33 卷 第 8 期 计 算 机 工 程 2007 年 4 月
Vol.33 No.8 Computer Engineering April 2007
·博士论文· 文章编号：1000—3428(2007)08—0023—04 文献标识码：A 中图分类号：TP393

基于免疫 Multi-agent 的网格入侵检测模型

倪建成[1,2]，李志蜀[1]，孙飞显[1]，梁 刚[1]，陈良银[1]
(1. 四川大学计算机学院，成都 610065；2. 曲阜师范大学计算机学院，日照 276826)

摘 要： 针对传统入侵检测技术难以适应动态的网格计算环境等问题，依据免疫原理，提出了一种基于 Multi-agent 的网格入侵检测模型(GIDIA)。描述了 GIDIA 的体系架构，给出了免疫模型、检测 Agent、决策 Agent 和防御 Agent 的定义，建立了相应的抽象数学模型及推理方程。理论分析和仿真结果表明，GIDIA 解决了信任社区内与社区间的协同预警及防御问题，具有检测率高、自适应能力强等特点，为实现网格安全提供了一种新方法。

关键词：网格安全；入侵检测；免疫性；Agent

Grid Intrusion Detection Model Based on Immune Multi-agent

NI Jiancheng[1,2], LI Zhishu[1], SUN Feixian[1], LIANG Gang[1], CHEN Liangyin[1]
(1. School of Computer, Sichuan Univ., Chengdu 610065; 2. School of Computer, Qufu Normal Univ., Rizhao 276826)

【Abstract】 Being that conventional intrusion detection systems can not adapt to the dynamic grid environment, grid intrusion detection model (GIDIA) based on application of immunity and multi-agent is proposed. In succession to describe the architecture, definitions of immune model, detective agent, decision-making and preventive agent are given. Relevant abstract mathematical models and detailed inferential equations are founded respectively. Theoretical analysis and experimental results show that GIDIA enables member sites in the same trust community or different ones to forewarn attacks cooperatively, and possesses higher detection rate a with better self-adaptability. GIDIA provides a way for implementation of grid security.

【Key words】 Grid security; Intrusion detection; Immunity; Agent

网格计算环境使复杂任务的协同解决成为可能，但网格安全问题依然严峻。网格安全基础设施 GSI 虽在一定程度上保证了资源的完整性、私密性和可用性，但难以阻止黑客入侵，可见网格入侵检测技术研究具有重要意义。

Choon等人[1]提出的网格入侵检测系统采用集中式方法分析和存储审计数据，存在单点失效、缩放性差等问题。Tolba等人[2]提出的分布式GIDA模型虽解决了缩放性问题，但无法防范黑客的刺探行为及分布式攻击。Hwang等人[3]提出的CAIDS系统中，基于审计日志的规则集缺乏及时性、完备性和动态适应能力，对未知攻击和资源滥用行为的检测效果较差。探索新的网格入侵检测技术十分必要。

生物免疫系统本身是一个复杂的自治Agent系统，其分布式、自适应、自学习和多样性等优良特征已引起了国内外学者的广泛关注，基于免疫原理的信息安全技术研究已取得了一定成果[4]。Agent的自治性、移动性、智能性和轻型特征使其成为检测入侵的理想载体[5]。

1 模型架构

GIDIA 模型包含以 GSI 组件为基础的安全物理通道层及由成员结点和信任社区构建的逻辑安全层，模型的体系架构如图 1 所示。GIDIA 的物理通道层通过认证、授权、代理、加密、作业调度、资源发现等措施保障 Agents 间的通信安全，完成检测任务的调度和分派，并利用动态策略管理机制实现策略协商和角色映射；逻辑安全层中的各信任社区通过融合各成员结点的决策信息形成疫苗，并利用疫苗检测入侵。检测 Agent(DetA)、决策 Agent(DecA)、防御 Agent(PreA)和控制 Agent(ConA)分别完成入侵检测、决策、防御和控制等任务。

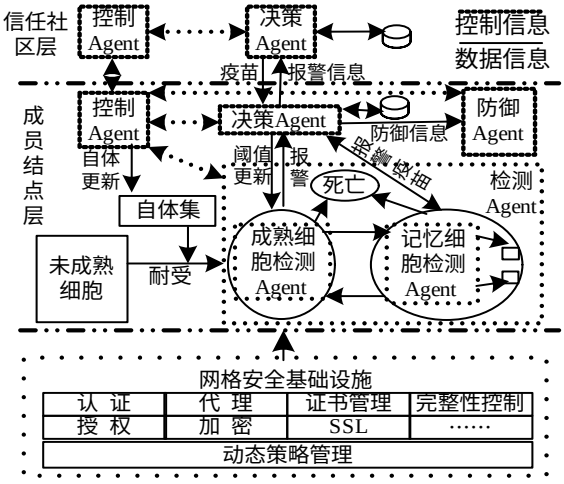

图 1 GIDIA 模型体系架构

在图 1 中，成员结点层基于免疫的DetA分为成熟细胞检测Agent($DetA_{Ma}$)和记忆细胞检测Agent($DetA_{Me}$)，负责在用户、系统、进程和网络等 4 个层次对系统进行动态免疫监视，

基金项目：国家自然科学基金资助项目（60072014）；四川省科技攻关项目（05GG021-003-2）；山东省自然科学基金资助项目（Q99G03）
作者简介：倪建成(1971－)，男，副教授、博士生，主研方向：网格安全；李志蜀，教授、博导；孙飞显、梁 刚，博士生；陈良银，副教授、博士生
收稿日期：2006-12-05 **E-mail：**nijch@163.com

若发现异常行为则向结点DecA报警；DecA负责处理报警信息和接收疫苗；PreA依据$DecA_{Me}$的决策信息采取相应的防御措施，例如在网络层采取断掉网络连接、改变防火墙规则，在进程层杀死可疑进程等；ConA负责刺激应答、更新自体集、管理和控制Agents以及提供用户界面等任务。信任社区层的Agents主要完成社区内或社区间的疫苗分发、报警信息聚类和提供用户界面等任务。社区数据库主要存储各结点的报警信息和社区管理策略。

2 模型理论

定义 1 免疫模型 $IM:=<U_{\Sigma}^{L},\psi,D_{\Sigma}^{L},Match,\theta,\omega,F>$。其中 U_{Σ}^{L} 为定义在有限字母表 $\Sigma=\{0,1\}$ 上的所有长度为L的字符串集，检测器集 $D_{\Sigma}^{L}\subseteq U_{\Sigma}^{L}$。设 Ω 为结点用户层、系统层、进程层和网络层的信息集，$\psi:\Omega\to U_{\Sigma}^{L}$ 为抗原提呈函数，令ad_i (i=1,...,4) 分别为上述 4 层信息的编码片段，则(ad_1,ad_2,ad_3,ad_4)为抗原决定基。*Match*为字符串匹配函数，θ为匹配数阈值，ω为敏感性阈值，F为亲和力检测函数。

定义抗原集合 $A=\{<a,ad>\mid a\in\Omega,ad\in U_{\Sigma}^{L},ad=\psi(a)\}$，则有 $\sum_{i=1}^{4}|ad_i|=L$；定义自体为正常网格服务存取行为及系统状态，否则为非自体。对自体集合 U_S 与非自体集合 U_N，有 $U_N\cup U_S=A$ 且 $U_N\cap U_S=\Phi$。

设 $X=\psi(X'),Y=\psi(Y'),\ X',Y'\in\Omega$，$\beta_i(i=1..4)$ 为 $X[ad_i]$ 与 $Y[ad_i]$(“ [ ] ”表示子串)间的亲和力匹配阈值，R为Rogers函数，则有

$$\mathrm{Match}(X,Y)=\begin{cases}1 & \text{iff } \exists i=1,...,4,\ \mathrm{s.t.}\ F(X[ad_i],Y[ad_i],\beta_i)=1\\ 0 & \text{otherwise}\end{cases} \tag{1}$$

$$F(X[ad_i],Y[ad_i],\beta_i)=\begin{cases}R(X[ad_i],Y[ad_i])= & \\ \dfrac{\sum_{j=1}^{|X[ad_i]|}\frac{}{(X[ad_i])_j\oplus(Y[ad_i])_j}}{\sum_{j=1}^{|X[ad_i]|}\frac{}{(X[ad_i])_j\oplus(Y[ad_i])_j}+2\sum_{j=1}^{|X[ad_i]|}(X[ad_i])_j\oplus(Y[ad_i])_j}\ge\beta_i & 1\\ \text{otherwise} & 0\end{cases} \tag{2}$$

定义 2 检测 Agent: *DetA* 为所有成熟细胞和记忆细胞的集合。即

$$\mathrm{DetA}=\{<g,gen,hc,hl[4]>\mid g\in D_{\Sigma}^{L},gen,hc\in\aleph,hl[i]\in\Re\ \ i=1,...,4\}$$

其中，g 为抗体基因片段，*gen* 为细胞代数，*hc* 为抗体与抗原的匹配数，*hl* 为累积亲和力数组，$\aleph$为自然数，$\Re$为实数。

设 LC 为成熟细胞生命周期，成熟细胞检测 Agent 表示为

$$\mathrm{DetA_{Ma}}=\{X\mid X\in\mathrm{DetA},\forall Y\in U_S,\mathrm{s.t.Match}(X.g,Y.ad)=0,X.hc<\theta,X.gen<LC\}$$

记忆细胞检测 Agent 表示为

$DetA_{Me}=\{X\mid X\in DetA,\forall Y\in U_S,\ s.t.$ $Match\ (X.g,Y.ad)=0,X.hc\ge\theta\}$。由定义 2 知

$$DetA_{Ma}\cup DetA_{Me}=DetA\qquad DetA_{Ma}\cap DetA_{Me}=\phi \tag{3}$$

定义 3 决策 Agent：令 cid 为攻击类型，el 为风险水平，则 $DecA:=\{<g,cid,el>\mid g\in D_{\Sigma}^{L},cid\in\aleph,el\in\Re\}$。

定义 4 防御 Agent: $\mathrm{Pre}A:=\{<g,cid>\mid g\in D_{\Sigma}^{L},cid\in\aleph\}$

因为网格服务的增加与删除及系统状态变迁会导致误报率升高，所以引入了动态自体集以适应动态变化的网格环境，其动态演化关系为

$$U_S(t)=\begin{cases}\{e\mid e\ is\ initial\ \ self\ \ elements\ ,e\in U_{\Sigma}^{L}\}\ t=0\\ U_S(t-1)-U_{Invalid}(t)\cup U_{New}(t)\qquad t\ge1\end{cases} \tag{4}$$

$$f_{Costimulat\ ion}(e)=\begin{cases}0\ ConA\ \ defines\ \ e\ as\ self\\ 1\ \ otherwise\end{cases} \tag{5}$$

$$U_{Invalid}(t)=\{e\mid e\in U_S(t-1),\exists d\in DetA(t)\ s.t.Match(e.ad,d.g)=1\wedge f_{Costimulation}(e)=1\}\cup\{e\mid e\ is\ deleted\ by\ ConA\} \tag{6}$$

其中，$U_{Invalid}(t)$为在t时刻是非自体而在t-1 时刻是自体的字符串集；$U_{New}(t)$为ConA新添加的自体集。

2.1 检测 Agent 模型

2.1.1 成熟细胞检测 Agent

$$\mathrm{DetA_{Ma}^{ing}(t)=\{X\mid Z\in DetA_{Ma}^{i}(t-1),\forall Y\in A(t)\ s.t.Match(Y.ad,Z.g)=0,X.gen=Z.gen+1,X.g=Z.g,X.hc=Z.hc,X.hl[j]=Z.hl[j]\ j=1,...,4\}} \tag{7}$$

$$DetA_{Ma}^{inh}(t)=\{X\mid Z\in DetA_{Ma}^{i}(t-1),(\exists Y\in A(t)\ s.t.Match(Y.ad,Z.g)=1\wedge f_{Costimulation}(Y)=1),X.gen=Z.gen+1,X.g=Z.g,X.hc=Z.hc+1,X.hl[j]=Z.hl[j]+R(Z.g[a_j],Y.ad[a_j])_{Max}\} \tag{8}$$

$$\mathrm{R(Z.g[a_i],Y.ad[a_i])_{Max}=max\{R(Z.g[a_j],Y.ad[a_j])\ \ j=1,...,4\}} \tag{9}$$

$$DetA_{Ma}^{id}(t)=\{X\mid X\in DetA_{Ma}^{i}(t-1),X.gen>LC,X.hc<\theta\}\cup\{X\mid X\in DetA_{Ma}^{i}(t-1),(\exists e\in U_{New}(t)\ s.t.Match(e.ad,X.g)=1)\} \tag{10}$$

$$DetA_{Ma}^{ia}(t)=\{X\mid X\in DetA_{Ma}^{inh}(t),\ X.hc\ge\theta-\omega\} \tag{11}$$

$$\mathrm{DetA_{Ma}^{cv}(t)=\{X\mid X\in\bigcup_{i}DetA_{Ma}^{ia}(t)\cup\bigcup_{j}DetA_{Me}^{jc}\ \bigcup_{k=1}^{\lambda}Y_k^{'},Y_k^{'}.gen=0,Y_k^{'}.hc=0,Y_k^{'}.hl[i]=0\ i=1..4,Y_k^{'}.g=Y.g^{'},Y.g^{'}\ne Y.g,Match(Y.g^{'},Y.g)=1,(\forall e\in U_S(t)\ s.t.Match(Y.g^{'},e.ad)=0)\}} \tag{12}$$

$$DetA_{Ma}^{i}(t)=\begin{cases}\{X_1,\ldots,X_n\} & t=0\\ DetA_{Ma}^{ing}(t)\cup DetA_{Ma}^{inh}(t)-DetA_{Ma}^{id}(t)-DetA_{Ma}^{ia}(t) & t\ge1\end{cases} \tag{13}$$

$$\mathrm{Det_{Ma}^{new}(t)=\{X\mid Y\ evolves\ from\ immature\ cells\ through\ self\ toleration,X.g=Y.g,X.gen=0,X.hc=0,X.hl[j]=0,j=1,...,4\}} \tag{14}$$

$$\mathrm{DetA_{Ma}}=\begin{cases}\phi & t=0\\ (\bigcup_{i}\mathrm{DetA_{Ma}^{i}(t)})\cup\mathrm{DetA_{Ma}^{cv}(t)}\cup\mathrm{DetA_{Ma}^{new}(t)} & t\ge1\end{cases} \tag{15}$$

$$\mathrm{DetA_{Ma}^{alarm}(t)=\{Y\mid Y\in DecA_{Me}(t),X\in\bigcup_{i}DetA_{Ma}^{ia}(t),Y.g=X.g,Y.el=X.hc,Y.cid=j\wedge X.hl[j]=max\{X.hl[k]k=1,...,4\}\}} \tag{16}$$

式(13)描述了第i个$\mathrm{DetA_{Ma}}$的演化趋势。在t时刻，未与任一抗原匹配而只进化一代的成熟细胞集由 $DetA_{Ma}^{ing}(t)$ 表示，否则，累积亲和力且进化一代的成熟细胞集由 $DetA_{Ma}^{inh}(t)$ 表示。式(10)表示如果成熟细胞超过生命周期而未累积足够的亲和力或与新加入的任何自体相匹配则将程序性死亡；式(11)表示在生命周期内因亲和力成熟被激活的免疫细胞集。

式(15)描述了所有成熟细胞的演化趋势。为应对猛烈攻击，系统将克隆被激活的成熟细胞和记忆细胞(式(12))，其中，克隆系数 λ 与攻击强度 ρ 成正比，与相类似的抗体数 S 成反比，即 $\lambda=C\rho/S$。

在 t 时刻，新产生的成熟细胞集由 $Det_{Ma}^{new}(t)$ 表示，而 $DetA_{Ma}^{alarm}(t)$ 为所有成熟细胞检测 Agent 发送给本地决策 Agent 的报警集合。

2.1.2 记忆细胞检测 Agent

由于成熟细胞完成的首次响应需要较长的学习期，不利于应对猛烈攻击，因此由记忆细胞完成的 2 次响应利用先验知识能够快速识别入侵。

$$DetA_{Me}^{ic}(t)=\{X\mid X\in DetA_{Me}^{i}(t-1),\exists Y\in A(t-1)\ s.t.Match(X.g,Y.ad)=1\wedge f_{Costimulation}(Y)=1\} \tag{17}$$

$$\mathrm{DetA}_{\mathrm{Me}}^{\mathrm{ing}}(\mathrm{t})=\{\mathrm{X}\mid \mathrm{Z}\in(\mathrm{DetA}_{\mathrm{Me}}^{\mathrm{i}}(\mathrm{t}-1)-\mathrm{DetA}_{\mathrm{Me}}^{\mathrm{ic}}(\mathrm{t})),\forall \mathrm{Y}\in \mathrm{A}(\mathrm{t})\ \mathrm{s.t.Match}(\mathrm{Z.g},\mathrm{Y.ad})=0,\mathrm{X.g}=\mathrm{Z.g},\mathrm{X.gen}=\mathrm{Z.gen}+1,\mathrm{X.hc}=\mathrm{Z.hc},\mathrm{X.hl}[\mathrm{j}]=\mathrm{Z.hl}[\mathrm{j}]\ \mathrm{j}=1,\ldots,4\} \tag{18}$$

$$\mathrm{DetA}_{\mathrm{Me}}^{\mathrm{inh}}(\mathrm{t})=\{\mathrm{X}\mid \mathrm{Z}\in\mathrm{DetA}_{\mathrm{Me}}^{\mathrm{i}}(\mathrm{t}-1),\exists \mathrm{Y}\in \mathrm{A}(\mathrm{t}-1)\mathrm{s.t.Match}(\mathrm{Z.g},\mathrm{Y.ad})=1\wedge \mathrm{f}_{\mathrm{Costimulation}}(\mathrm{Y})=1,\mathrm{X.g}=\mathrm{Z.g},\mathrm{X.gen}=\mathrm{Z.gen}+1,\mathrm{X.hc}=\mathrm{Z.hc}+1,\mathrm{X.hl}[\mathrm{j}]=\mathrm{Z.hl}[\mathrm{j}]+\mathrm{R}(\mathrm{Z.g}[\mathrm{a_j}],\mathrm{Y.ad}[\mathrm{a_j}])_{\max}\ \mathrm{j}=1,\ldots,4\} \tag{19}$$

$$DetA_{Me}^{id}(t)=\{X\mid X\in(DetA_{Me}^{i}(t-1)-DetA_{Me}^{ic}(t)),\exists Y\in U_S(t)\,s.t.Match(X.g,Y.ad)=1\wedge f_{Costimulation}(Y)=0\} \tag{20}$$

$$DetA_{Me}^{i}(t)=\begin{cases}\{X_1,\ldots,X_n\} & t=0\\ DetA_{Me}^{ing}(t)\cup DetA_{Me}^{inh}(t)-DetA_{Me}^{id}(t) & t\geq 1\end{cases} \tag{21}$$

$$\mathrm{DetA}_{\mathrm{Me}}^{\mathrm{new}}(\mathrm{t})=\{\mathrm{X}\mid \mathrm{Y}\in\bigcup_{\mathrm{i}}\mathrm{DetA}_{\mathrm{Ma}}^{\mathrm{ia}}(\mathrm{t}),\mathrm{X.g}=\mathrm{Y.g},\mathrm{X.gen}=0,\mathrm{X.hc}=\mathrm{Y.hc},\mathrm{X.hl}[\mathrm{j}]=\mathrm{Y.hl}[\mathrm{j}]\ \mathrm{j}=1,\ldots,4\} \tag{22}$$

$$DetA_{Me}^{tcv}(t)=\{X\mid Y=DecA^{tcv}(t),(\forall e\in U_S(t)\,s.t.Match(Y.g,e.ad)=0)\} \tag{23}$$

$$DetA_{Me}(t)=\begin{cases}\{X_1,\ldots,X_n\} & t=0\\ (\bigcup_i DetA_{Me}^{i}(t))\cup DetA_{Me}^{new}(t)\cup DetA_{Me}^{tcv}(t) & t\geq 1\end{cases} \tag{24}$$

$$\mathrm{DetA}_{\mathrm{Me}}^{\mathrm{alarm}}(\mathrm{t})=\{\mathrm{Y}\mid \mathrm{Y}\in\mathrm{DecA}(\mathrm{t}),\mathrm{X}\in\bigcup_{\mathrm{i}}\mathrm{DetA}_{\mathrm{Me}}^{\mathrm{ic}}(\mathrm{t}),\mathrm{Y.g}=\mathrm{X.g},\mathrm{Y.el}=\mathrm{X.hc},\mathrm{Y.cid}=\mathrm{j}\wedge\mathrm{X.hl}[\mathrm{j}]=\max\{\mathrm{X.hl}[\mathrm{k}]\mathrm{k}=1,\ldots,4\}\} \tag{25}$$

式(21)描述了第 i 个记忆检测 Agent 的动态演化关系。由 $DetA_{Me}^{ing}(t)$ 表示的记忆细胞因未匹配任何抗原而只增加年龄，由 $DetA_{Me}^{inh}(t)$ 表示的记忆细胞则累积亲和力并进化一代，且与非自体抗原匹配的记忆细胞将被克隆(式(12)、式(17))。为避免误报，与自体匹配的记忆细胞则应死亡(式(20))。

式(24)给出了所有记忆细胞的演化趋势。其中，$DetA_{Me}^{new}(t)$ 包含了所有成熟细胞检测Agent中因亲和力成熟转变而来的记忆细胞。当$DetA_{Me}$检测到入侵时，如式(25)所示，它将向结点决策Agent发出相应的报警信息。

值得注意的是，自体集因异质结点提供的服务和采取的管理策略不尽相同而有所差异。如式(23)所示，为了降低误报率，来自其它结点的疫苗首先进行自体耐受，只有当疫苗不与该结点的任何自体匹配时才能成为记忆细胞。

2.2 决策 Agent 模型

2.2.1 社区决策 Agent

社区决策 Agent 把成员结点的决策信息按相似性分为不同家族，并形成疫苗在社区内或社区间依策略进行分发。

$$DecA_{Tc}^{new}(t)=\{X\mid X\in\bigcup_i DecA_{Me}^{ialarm}(t)\} \tag{26}$$

$$Family(X)=\{Y\mid Y\in(DecA_{Tc}(t-1)\cup DecA_{Tc}^{new}(t))\,s.t.Match(X.g,Y.g)=1\} \tag{27}$$

$$Family_{Det}(t)=\{e_i\mid e_i\in(DecA_{Tc}(t-1)\cup DecA_{Tc}^{new}(t))=\bigcup_{i=1}^{K}Family(e_i)\wedge(\forall j,i\neq j,Match(e_i.g,e_j.g)=0)\} \tag{28}$$

$$DecA_{Tc}^{tcv}(t)=\{e_i\mid e_i\in Family_{Det}(t),\sum_{j=1}^{|Family(e_i)|}(X_j.el)\geq C_1\theta,X_j\in Family(e_i)\} \tag{29}$$

$$DecA^{tcv}(t)=\{Y\mid X\in DecA_{Tc}^{tcv}(t),Y\in DetA,Y.g=X.g,Y.gen=0,Y.hc=0,Y.hl[j]=0\ j=1..4\} \tag{30}$$

$$DecA_{Tc}^{d}(t)=\{Y\mid Y\in Family(X),X\in DecA_{Tc}^{tcv}(t)\} \tag{31}$$

$$DecA_{Tc}(t)=\begin{cases}\phi & t=0\\ DecA_{Tc}(t-1)\cup DecA_{Tc}^{new}(t)-DecA_{Tc}^{d}(t) & t\geq 1\end{cases} \tag{32}$$

式(32)给出了社区决策Agent的动态演化趋势。在t时刻，$DecA_{Tc}^{new}(t)$ 表示了来自结点决策Agent的所有决策信息。式(27)对社区拥有的决策信息进行分类，然后形成由 $Family_{Det}(t)$ 表示的代表元素集。如果代表元素集所在类的所有元素的风险之和超出阈值$C_1\theta$(式(29))，那么代表元将作为疫苗分发至成员结点(式(30)、式(23))，代表元所在类的元素将被存入全局数据库(式(31))。

2.2.2 成员结点决策 Agent

结点决策 Agent 将免疫 Agents 的报警信息进行分类并形成防御和决策信息，分别发送至防御 Agent 和社区决策 Agent。此外，它负责向检测 Agent 传递疫苗和风险阈值。

$$DecA_{Me}(t)=\begin{cases}\phi & t=0\\ DecA_{Me}(t-1)\cup DetA_{Ma}^{alarm}(t)\cup DetA_{Me}^{alarm}(t)-DecA_{Me}^{d}(t) & t\geq 1\end{cases} \tag{33}$$

$$Family'(X)=\{Y\mid Y\in(DecA_{Me}(t-1)\cup DetA_{Ma}^{alarm}(t)\cup DetA_{Me}^{alarm}(t))\,s.t.Match(X.g,Y.g)=1\} \tag{34}$$

$$Family'_{Det}(t)=\{e_i\mid e_i\in(DecA_{Me}(t-1)\cup DetA_{Ma}^{alarm}(t)\cup DetA_{Me}^{alarm}(t))=\bigcup_{i=1}^{K}Family(e_i)\wedge(\forall j,i\neq j,Match(e_i.g,e_j.g)=0)\} \tag{35}$$

$$DecA_{Me}^{mev}(t)=\{e_i\mid e_i\in Family'_{Det}(t),\sum_{j=1}^{|Family'(e_i)|}(X_j.el)\geq C_2\theta,X_j\in Family'(e_i)\} \tag{36}$$

$$DecA_{Me}^{ialarm}(t)=\{Y\mid X\in DecA_{Me}^{mev}(t),Y.g=X.g,Y.el=\sum_{j=1}^{|Family'(X)|}(Z_j.el),Z_j\in Family'(X),Y.cid=X.cid\} \tag{37}$$

$$DecA_{Me}^{d}(t)=\{Y\mid Y\in Family'(X),X\in DecA_{Me}^{mev}(t)\} \tag{38}$$

$$DecA_{Me}^{pre}(t)=\{Y\mid Y\in \mathrm{Pre}A,X\in DecA_{Me}^{mev}(t),Y.g=X.g,Y.cid=X.cid\} \tag{39}$$

式(33)反映了成员结点决策Agent的动态演化趋势。首先，来自检测Agent的报警信息被分类(式(34))并形成由 $Family'_{Det}(t)$ 表示的代表元素集。如果代表元所在类的报警威胁程度之和超过阈值$C_2\theta$(式(36))，那么，代表元将被作为疫苗(集合为 $DecA_{Me}^{ialarm}(t)$)传递给社区决策Agent，同时根据攻击分类属性形成防御信息发送至防御Agent(式(39))。式(38)描述了存储至局部数据库的相应信息。

2.3 防御 Agent 模型

$$\mathrm{Pre}A(t)=\begin{cases}\phi & t=0\\ \{Y\mid X\in DecA_{Me}^{pre}(t),Y.g=X.g,Y.cid=X.cid\} & t\geq 1\end{cases} \tag{40}$$

当防御 Agent 接收到防御信息时，依据攻击分类属性在 4 个不同层次(用户层 cid=1，系统层 cid=2，进程层 cid=3，网络层 cid=4)上采取相应防御措施。

3 仿真实验

3.1 实验平台与参数设置

为验证模型的有效性，在模拟网格环境下进行了实验。实验平台由 20 台工作站、1 台服务器组成，利用网格模拟软件 GridSim 4.0 将硬件环境仿真为由 3 个信任社区组成的拓扑结构：TC1、TC2 各包含 4 台工作站(配置相同但提供的服务不同)；TC3 包含 12 台工作站、1 台服务器。

实验参数分别设置为：克隆系数λ=1，激活阈值θ=10，字符串匹配阈值β_i = 0.6(i=1,...,4)，生命周期LC=5，ω初始值为 0，L=512。为简化实验，置C_1=1，即社区$DecA$接收到任一成员结点的报警则进行疫苗分发；C_2=1，即结点$DecA$检测到报警

则立即采取防御措施并向社区Agent报警。

3.2 实验结果与分析

为验证对接种疫苗进行自体耐受的必要性，利用Ftp-write等工具攻击社区1和社区2内的所有机器。

在社区1，疫苗未进行自体耐受直接成为记忆细胞，相反，在社区2，疫苗要首先进行自体耐受，耐受成功后才能成为记忆细胞。社区1的平均误报率为0.18(最高为0.34，最低为0.03)，社区2的平均误报率为0.04(最高为0.09，最低为0.03)。结果表明：疫苗经历自体耐受再检测入侵的情况下，FP值平均下降了0.14。这也说明由于疫苗会对不同结点的部分自体产生自我免疫反应，因此成为提高系统误报率的潜在因素。

为验证和度量本模型的有效性，对TC3进行Udpstorm、Land、Ping-of-Death等19种攻击。定义ROC量度(即正确检测率和误报率的对应关系)为

$$ROC := \{< TP, FP > | \, TP \text{ is True Positive Rate}, \; FP \text{ is False Positive Rate}\} \tag{41}$$

4类典型攻击的ROC曲线如图2所示。社区对系统TP的影响对比如图3所示。

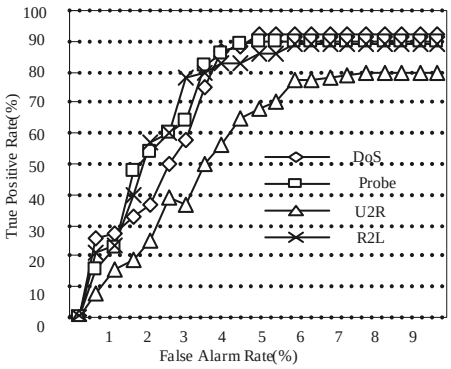

图2 4类攻击对应的ROC曲线

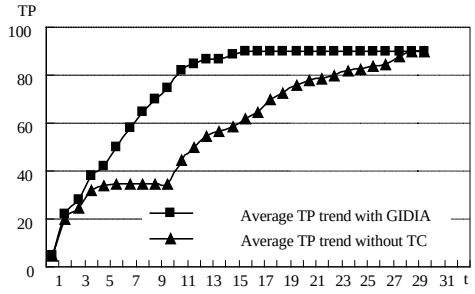

图3 社区对系统TP的影响对比

由图2可以看出，伴随FP的少量递增，TP会快速地达到并保持一种近似稳定状态。DoS攻击具有最高92%的正确检测率和5%的误报率；Probe和R2L攻击具有基本相同的TP值，分别为90%与89%，但它们必须同时忍受5%与6%的误报率；为了获得高于80%的正确检测率，U2R至少具有8%的误报率。

为了验证社区疫苗对检测速度的影响，类似于实验2对社区3先后共进行了相似的19种攻击。结果如图3所示。可以看出，当采用GIDIA模型在整个社区内协同防御入侵时，系统在时刻16达到TP的最高平均值89%，既而保持稳定状态。但是，在没有加入社区的单机系统中，TP直至时刻29才达到最高平均值89%。结果说明协同防御对加快入侵检测系统的响应速度和提高检测率是非常重要的因素之一。

4 结论

GIDIA采用的3层逻辑架构充分利用了成员结点用户层、系统层、进程层和网络层的多层次信息检测入侵，通过信任社区的疫苗分发使成员结点在未受攻击时提前预警，提高了2次响应的速度，对接种疫苗进行自体耐受降低了误报率。同时，GIDIA良好的可缩放性和健壮性可以实现信任社区之间的预警、检测和防御功能，各类Agent的协同工作和动态演化机制能更好地适应动态的网格计算环境。然而，GIDIA模型在网格安全风险评估及动态取证等方面还存在一定的不足，这将是未来的研究重点。

参考文献

1 Choon O T, Samsudim A. A Grid-based Intrusion Detection System[C]//Proc. of the 9th IEEE Asia-pacific Conference on Communications. 2003: 1028-1032.

2 Tolba M F. GIDA: Toward Enabling Grid Intrusion Detection Systems[C]//Proc. of the 5th IEEE International Symposium on Cluster Computing and the Grid. 2005.

3 Hwang K. GridSec: Trusted Grid Computing with Security Binding and Self-defense Against Network Worms and Ddos Attacks[C]//Proc. of the International Workshop on Grid Computing Security and Resource Management. 2005: 187-195.

4 Graaff A J, Engelbrecht A P. Optimised Coverage of Non-self with Evolved Lymphocytes in an Artificial Immune System[J]. International Journal of Computational Intelligence Research, 2006, 2(2): 127-150.

5 Hegazy I M, Faheem H M. Evaluating How Well Agent-based IDS Performe[J]. IEEE Potentials, 2005, 24(2): 27-30.

(上接第22页)

参考文献

1 Pasquier N, Bastide Y, Taouil R, et al. Discovering Frequent Closed Itemsets for Association Rules[C]//Proc. of the 7th Int. Conf. on Database Theory, Jerusalem, Israel. 1999-01: 398-416.

2 Han J, Pei I, Yin Y. Mining Frequent Patterns Without Candidate Generation[C]//Proc. of SIGMOD'00, Dallas. 2000: 1-12.

3 Han Jiawei, Kamber M. 数据挖掘: 概念与技术[M]. 范 明, 孟小峰, 译. 北京: 机械工业出版社, 2001: 149-184.

4 Pei J, Han J, Wang J. Closet+: Searching for the Best Strategies for Mining Frequent Closed Itemsets[C]//Proc. of SIGKDD'03, Washington, D. C.. 2003-08: 236-245.