分散型侵入検知システムにおける免疫型診断モデルを用いた自己監視

豊橋技術科学大学　○渡邊 裕司　石田 好輝

Self-monitoring with Immunity-Based Diagnostic Model in Distributed Intrusion Detection System

○ Yuji WATANABE and Yoshiteru ISHIDA, Toyohashi University of Technology

Abstract: In distributed intrusion detection system (DIDS), self-monitoring can be a difficult problem. One possibility is that each IDS is checked periodically by others. In this paper, we propose mutual tests among IDSs using immunity-based diagnosis. Some simulation results show that corrupted IDS gradually decreases its credibility. Furthermore, we compare self-monitoring using direct communication between IDSs with one using mobile agent.

Key Words: Intrusion Detection System, Immunity-based diagnosis, Self-monitoring, Mobile agent

1　はじめに

侵入検知システム（Intrusion Detection System，以下IDSと略す）とは，コンピュータおよびネットワークに対するセキュリティ侵害の検出，通知，検出情報の管理を行うシステムである．そのタイプとして，ホスト型とネットワーク型，不正検出と異常検出，集中型と分散型などに分類される．最近では，単独のIDSだけではなく，ネットワークに分散して配備された複数のIDSを連携させることの必要性が唱えられている[1),2)]．

IDS構築の一つのアプローチとして，生物の防御機構である免疫系から着想を得ることは有益であり，すでに幾つかの研究が存在する．Forrest ら[3)]は，免疫系の自己・非自己の識別方法をもとに，Unixプロセスに対して自己（正常なふるまい）を定義し，実験により異常なふるまいを検出できることを示した．さらに，胸腺におけるT細胞の選択（negative selection）を参考にした検出器の生成法[4)]や，その他の免疫特性（多様性，分散検出，記憶など）も取り入れたARTISフレームを適用したネットワーク型IDS（LISYS）[5)]を提案した．また，溝口ら[6)]による免疫細胞間の協調に基づいた不正侵入とウィルス感染に対処する方法もある．免疫細胞をエージェントとみなすのならば，Crosbie，Spafford ら[7),8)]による自律エージェントを用いたIDSも挙げられ，彼らの論文[7)]の中でも免疫系との類似性が指摘されている．さらに，生体の免疫細胞が体内を移動（循環）していることから，エージェントを移動させるアプローチ，すなわちモバイルエージェントを用いたシステムも存在する[9)–11)]．

これら複数のエージェントをネットワーク上に配置したIDSは並列分散型の範疇に入る．分散型IDSは，一枚岩な（monolithic）IDSと比較して，拡張性，耐障害性，動的な再構成などで優ると言われている[8)]．しかし，耐障害性に関して，侵入者によってエージェント（あるいはIDS）自身が改ざんされたり，エージェントの設計に誤りがあった場合を想定した研究や議論はまだ少ない．論文[5),8)]では，自己監視（self-monitoring）という用語を用いて，各エージェントが相互にチェックするという考えは示されているものの，詳細な検証は行われていない．IDSの改ざんは，正常なふるまいを異常とみなす誤報（false positive）や異常なふるまいを正常とみなす欠報（取りこぼし，false negative）を引き起こす．

そこで，生体の免疫系とその他の応用を眺め直してみると，B細胞間の相互作用を唱えた「免疫ネットワーク説[12)]」から発想を得た「免疫型診断モデル」が提案されている[13)–15)]．このモデルは，相互にテストし得るユニットからなるシステムに対して，そのテスト結果並びにユニットの活性・非活性をもとに，ダイナミカルモデルによって各ユニットの正常・異常を判定する．このモデルは，セメントプラントなどの診断に応用されていたが，最近，筆者ら[16)]によってモバイルエージェントシステムにおける誤動作ホスト・エージェントの検出に応用された．

本論文では，分散型IDSに対して免疫型診断モデルを用いて自己監視を行い，ルールが改ざんされたIDSを検出する．分散型IDSを単純化したシミュレーションシステムによって，免疫型診断モデルがどのようなパラメータ条件のもとで適切に検出できるかを調べる．さらに，文献[9)–11)]でモバイルエージェントを用いた方法が挙げられているものの，モバイルエージェントを用いる利点や欠点は十分に示されていない．そこで，IDS間で自己監視を行うにあたり，ホスト間通信を用いる場合とモバイルエージェントを用いる場合の二つの方法に対して，診断性能の比較検証を行う．

2　免疫型診断モデル[14),15)]

まず，Fig. 1の左に示す5個のユニットからなる相互評価のネットワークにおいて，信用できないユニット（抗原）を探すことを考える．ここで，ユニットiとjの双方向矢印が＋ならばお互いに信用できると評価し，逆に－ならば信用できないと考えているとする．

各ユニットで単純に正負の票を集票すると，ユニット2，3，5については同じ値となり，どれが信用できないか区別できない．しかし，信用できないユニットの評価結果も信用できないことから，それを重みづけする．ユニットに対する集票結果が正のときノードは活性（信用度が1），負のとき非活性（信用度が0）とする．はじめは全てのユニットが活性であるとし，同期的に全ユニットが評価結果にユニットの活性・非活性を重みづけして集票し，その集票結果から信用度を更新することを繰り返すと，信用度のベクトルはFig. 1の右に示すように変化する．つまりユニット4，5が信用できないことになる．

上記の離散的なモデルを連続系の動的モデルで捉え直すと，ユニットiの信用度R_iは，以下のダイナミカル

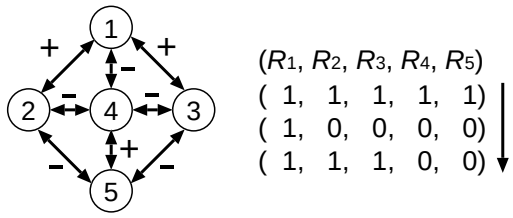

Fig. 1 An example of mutual evaluation network.

モデルによって変化する．

$$\frac{dr_i(t)}{dt} = \sum_j T_{ji} R_j + \sum_j T_{ij} R_j - \frac{1}{2} \sum_{j \in \{k:T_{ik} \neq 0\}} (T_{ij} + 1) \tag{1}$$

$$R_i(t) = \frac{1}{1 + \exp(-r_i(t))} \tag{2}$$

ここで，$r_i \in (-\infty, \infty)$ は媒介変数であり，式 (2) により信用度 $R_i \in [0, 1]$ へ変換される．また，T_{ij} は，ユニット i がユニット j をテストした結果であり，テストが正しければ 1，テストが間違っていれば -1，テストが存在しなければ 0 とする．具体的なテスト方法については **3.2** で述べる．

式 (1) の右辺の第 1 項は，ユニット i が他からの評価を総合している項であり，右辺第 2，3 項はユニット i が他を評価することにより，他から反射的に評価されているものを総合している項である．換言すれば，この項は，ある人が信用の高い人に対してうそつきだというと，逆にその人自身の信用が低く評価されてしまうことに対応する．なお，このモデルはホップフィールドネットワークと同様にある種のエネルギー関数を用いることにより収束性が保証されている[13]．

3 シミュレーションシステム

3.1 分散型 IDS

診断モデルの基本的なふるまいを調べるための，分散型 IDS を単純化したシミュレーションシステムについて説明する．IDS 間で自己監視をするにあたり，ホスト間通信を用いる場合とモバイルエージェントを用いる場合の二つの方法を取り上げる．

両方法に共通する項目として，第一に，Fig. 2 に示すように N 台の各ホスト上で IDS が起動しているとし，各ホストは $N_c (2 \leq N_c < N)$ 台のホストと「接続」することでネットワークを形成する．ここでの「接続」とは，他のホストを介さずに「直接」ルールやエージェントを送受信できることを意味する．

第二に，各 IDS が検査に用いるルールに関してである．例えば，フリーの IDS である snort[17] には，約 1500 個のルールがデフォルトでは/etc/snort/にサービス毎にファイルとしてまとめられており，ルールの記述は複雑である（Fig. 3）．さらに，運用する際にはセキュリティポリシーなどをもとにルールのチューニングが必要であり，ルール記述のバリエーションも増える．シミュレーションでは単純化を行い，ルールをラベル（A,B,...）とデータ（あるいは条件部と行動部）の対で表し，データは 0 と 1 のみをとるとする．また，ルールの総数を L とすると，一般に各ホストは異なるサービスを提供していることから，各 IDS が利用するルールにも相違があると仮定し，IDS が持つ平均ルール数を $L_h (\leq L)$ とする．つまり，$L_h = L$ ならば同一の IDS からなる均質なネットワークであり，$L_h \leq L/N$ ならば IDS が全く異なる不均質なネットワークとなる．

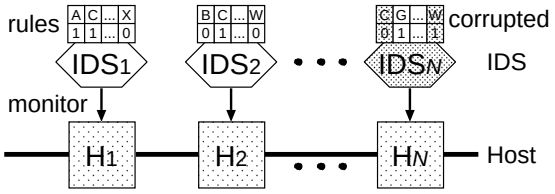

Fig. 2 Model of distributed intrusion detection system.

```
[root@wata root]# ls /etc/snort/
RCS                       local.rules         snmp.rules
attack-responses.rules    misc.rules          snort.conf
backdoor.rules            multimedia.rules    snort_tutkie.conf
bad-traffic.rules         mysql.rules         sql.rules
chat.rules                netbios.rules       telnet.rules
classification.config     nntp.rules          tftp.rules
ddos.rules                oracle.rules        virus.rules
deleted.rules             other-ids.rules     web-attacks.rules
dns.rules                 p2p.rules           web-cgi.rules
dos.rules                 policy.rules        web-client.rules
experimental.rules        pop3.rules          web-coldfusion.rules
exploit.rules             porn.rules          web-frontpage.rules
finger.rules              reference.config    web-iis.rules
ftp.rules                 rpc.rules           web-misc.rules
icmp-info.rules           rservices.rules     web-php.rules
icmp.rules                scan.rules          x11.rules
imap.rules                shellcode.rules
info.rules                smtp.rules
[root@wata root]# head /etc/snort/web-cgi.rules
# (C) Copyright 2001,2002, Martin Roesch, Brian Caswell, et al.
#    All rights reserved.
# $Id: web-cgi.rules,v 1.56 2002/08/18 20:28:43 cazz Exp $
#---------------
# WEB-CGI RULES
#---------------
#

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI
HyperSeek hsx.cgi directory traversal attempt"; uricontent:"/hsx.cgi";
content:"../../"; content:"%00"; flow:to_server,established;
reference:bugtraq,2314; reference:cve,CAN-2001-0253;
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI
HyperSeek hsx.cgi access"; uricontent:"/hsx.cgi"; flow:to_server,
established; reference:bugtraq,2314; reference:cve,CAN-2001-0253;
classtype:web-application-activity; sid:1607;  rev:3;)
[root@wata root]#
```

Fig. 3 Example of snort rules.[17]

最後に，正しいラベル・データ対があらかじめ決定されているもとで，改ざんルールをデータ部の 01 反転で表す．改ざんされた IDS は，所持する全ルールのうち割合 $e (0 < e \leq 1)$ で改ざんルールを含むものとする．IDS のルールに改ざんがなければ，IDS はルールに従ってホストの監査証跡（ログ）を検査し，ホストの異常の有無を判定できる．しかし，IDS のルールに改ざんがあると (例えば，Fig. 2 における IDS_N のラベル C や W のデータ)，ホストの正常なふるまいを異常とみなす誤報，異常なふるまいを正常とみなす欠報が発生する．免疫型診断モデルは，他のホスト上の IDS と相互にテストし信用度を更新することで，ルールが改ざんされた IDS を検出する．シミュレーションでは，すでに何らかの侵入が行われ，$N_f (\leq N)$ 個の IDS が改ざんされた状態から開始する．

3.2 ホスト間通信の場合

診断を行うためには，相互テストの方法つまり式 (1) の T_{ij} を決定する必要がある．本節ではホスト（IDS）間で直接通信する方法を示す．Fig. 4 に示す IDS_i と IDS_j の間のやりとりを通じて相互テストと信用度の更新を解

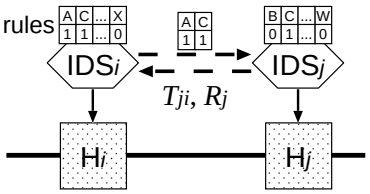

Fig. 4 Self-monitoring using direct communication between hosts.

説する．

1. IDS_i は，テスト用にランダムに選んだ $L_t(\leq L_h)$ 個のルールを IDS_j に送信する（Fig. 4 では $L_t = 2$）．
2. IDS_j は，受信したルールと所持するルールを比較し，対応するラベルがあるデータに対して，値の一致・不一致を調べ，テスト結果 T_{ji} を以下のように決定する．

$$T_{ji} = \begin{cases} 1 & \text{比較した全ルールが一致する} \\ -1 & \text{一つでも不一致のルールがある} \\ 0 & \text{比較できるルールがない} \\ -1/0/1 & \mathrm{IDS}_j \text{ は改ざんされている} \end{cases} \quad (3)$$

   Fig. 4 では，ラベル A のデータは所持していないため，ラベル C のデータだけをみて $T_{ji} = 1$ となる．
3. IDS_j は，テスト結果 T_{ji} と自分の信用度 R_j を IDS_i に送る．
4. 逆に，IDS_i による IDS_j のテストにより T_{ij} が求まる．
5. IDS_i は，接続している $N_c - 1$ 台の他のホスト上の IDS とも上記の処理を行った後，式 (1) (2) により信用度 R_i を更新する（他の IDS も同様）．
6. 以上の処理を一ステップとし，信用度の更新を繰り返す．

ここで正確な診断に至るために重要なパラメータとして，各 IDS がどれだけの IDS と「直接」通信してテストできるかを表す一ホストあたりの接続ホスト数 N_c が挙げられる．より多くの IDS とテストできればよいが，そうすると通信量が増加するため，できるだけ少ない接続ホスト数でも適切な検出ができることが望ましい．そこでシミュレーションでは，このパラメータを変化させて性能を評価する．なお，他のホストを介した「間接」通信もありえるが，これはテスト用ルールがホスト間を移動しているとみなせば，次節のモバイルエージェントの範疇に入ると考えられるため，ここでは直接通信に限るとする．また，通信量の観点からは，一ステップで送受信するテスト用ルール数 L_t も重要なパラメータであるが，ルール数が多ければ少ないステップ数で診断が収束するという診断速度との関連が強いため，シミュレーションでは固定値（ルール総数 L の 100 分の 1）とする．

また，検査される側の立場から被検査データをどれだけ所持するかを示す IDS が持つ平均ルール数 L_h も重要である．IDS があまりルールを所持していないと，同じルールを所持している IDS が少なくなり，正しい判定に至らない可能性があるからである．このパラメータも変化させてシミュレーションを行う．

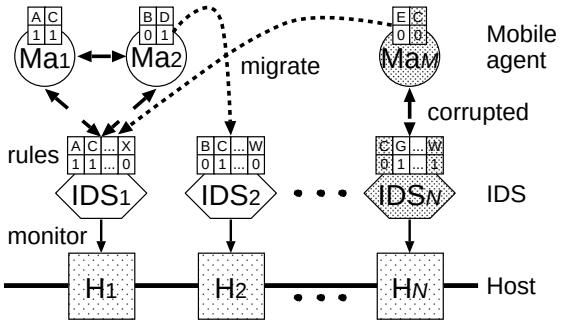

Fig. 5 Self-monitoring using mobile agents.

3.3 モバイルエージェントの場合

Fig. 5 に示すモバイルエージェントを用いた方法では，前節の送受信されるテスト用ルールに「移動」と「信用度評価」の機能を与えたものを，モバイルエージェントとみなす．そのモバイルエージェントが各 IDS と相互にテストする（エージェントの移動によりテストする IDS は絶えず変化する）．また，同じホスト上にいるエージェント同士でも相互にテストできるとする（Fig. 5 の Ma_1 と Ma_2）．ただし，異なるホスト上のエージェント間でもテストをすると，エージェントが移動することの意義がなくなるため行わない（Ma_1 と Ma_M）．なお，相互テストと信用度の更新は前節と同じとする．

前節のホスト間通信の方法と比較を行うために，エージェントが持つルール数は前節の送受信するテスト用ルール数 L_t と同じにして，一ホストあたりのエージェント数 M/N を変化させて性能評価を行う．つまり，前節で検査体の個数に対応する接続ホスト数 N_c の代わりに，エージェント数を変化させ，できるだけ少ない個数でも適切な検出ができるかどうかを調べる．なお，IDS が持つ平均ルール数 L_h に関してはモバイルエージェントの場合でも同じように変化させる．

最後に注意すべき点として，エージェントは最初に IDS のルールからランダムに選んだ L_t 個のルールを受け継いで生成され，Fig. 5 の Ma_M のように改ざんされたルールを持つエージェントも存在することである．これは，前節のホスト間通信の場合でも，送受信するテスト用ルールは IDS のルールから選ばれ，その中には改ざんルールも含まれることと同じである．

4 シミュレーション

4.1 評価指標

シミュレーションの各ステップでは，IDS（またはエージェント）の信用度 R_i を記録するとともに，「誤報（false positive）」と「欠報（false negative）」も求める．誤報とは「正常」にも関わらず「異常」と判定されることで，逆に欠報とは「異常」にも関わらず「正常」と判定されることであり，両者をできるだけ小さくすることが望まれる．本シミュレーションでは，誤報率 α と欠報率 β を

Table. 1 Parameters list.

説明	変数名	値
信用度の初期値	$R_i(0)$	1.0
媒介変数の初期値	$r_i(0)$	1.0
ホスト (IDS) の台数	N	50
改ざん IDS の台数	N_f	15
改ざんの割合	e	0.5
ルールの総数	L	5000
IDS が所持する平均ルール数	L_h	可変
送受信するルール数 ＝エージェントが持つルール数	L_t	50
一ホストあたりの接続ホスト数 （ホスト間通信の場合） （モバイルエージェントの場合）	N_c	 可変 5
一ホストあたりのエージェント数 （ホスト間通信の場合） （モバイルエージェントの場合）	M/N	 0 可変

以下ように定義する：

$$\alpha = \frac{N_t^{low}}{N - N_f} \tag{4}$$

$$\beta = \frac{N_f^{high}}{N_f} \tag{5}$$

ここで，$N - N_f$ は正常な IDS の個数，N_f は改ざん IDS の個数である．また，N_t^{low} は正常な IDS のうち信用度が 0.8 以下のものの個数を，N_f^{high} は改ざん IDS のうち信用度が 0.2 以上のものの個数を表す．

Table 1 に挙げるパラメータのもとで，Java で作成したシミュレータを用いて，一試行 30 ステップまで計算する．初期配置などを変えて 50 試行行い，平均誤報率 $\bar{\alpha}$ と平均欠報率 $\bar{\beta}$ を求める．Table 1 でパラメータの値が可変となっているものは，値を変更して検証することを示す．

4.2 信用度の推移による比較

パラメータを固定（$L_h = 500, N_c = M/N = 6$）して，信用度の推移を観察した．Fig. 6 にホスト間通信の場合，Fig. 7 にモバイルエージェントの場合の正常な IDS と改ざん IDS の信用度の変化を示す．ここでの信用度は，ある一つの IDS に着目して，試行を 50 回繰り返したときの平均となっている．両方とも正常な IDS は高い信用度に，改ざん IDS は低い信用度となり，改ざん IDS の検出ができていることが分かる．

両者を比較すると，ホスト間通信の方は正常な IDS の信用度が若干低く，モバイルエージェントの方は改ざん IDS の信用度減少に時間を要している．ホスト間通信で信用度が低い点は，正常な IDS にも関わらず信用度が減少する誤報が 50 試行のうち数回発生しているためであり，次節でより詳細に調査する．一方，モバイルエージェントの収束が遅い点は，まずエージェントが評価された後で IDS が評価されるため遅れが生じていると考えられる．診断速度に関しては，ホスト間通信に比べてモバイルエージェントはあまりよくないといえる．

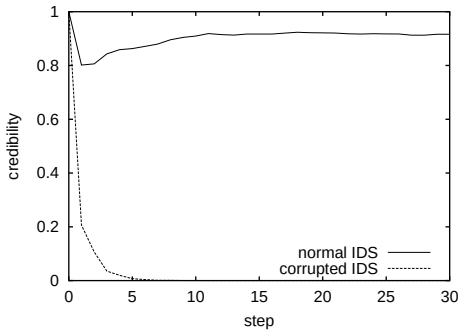

Fig. 6 Transitions of credibility for normal and corrupted IDS over 50 trials using direct communication between IDSs.

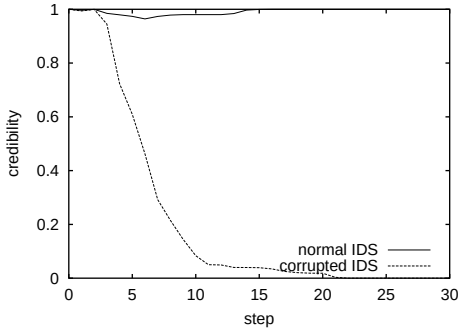

Fig. 7 Transitions of credibility for normal and corrupted IDS over 50 trials using mobile agent.

4.3 誤報・欠報による比較

3.2 で述べたように，ホスト間通信の場合，検査体がどれだけ存在するかを表す一ホストあたりの接続ホスト数 N_c と，非検査体がどれだけのルールを所持するかを表す IDS が所持する平均ルール数 L_h が，診断の成否を決める重要なパラメーラであると考えられる．

Fig. 8 は，N_c と L_h を変化させた場合の 30 ステップ後の平均誤報率 $\bar{\alpha}$ と平均欠報率 $\bar{\beta}$（50 試行の平均）を示す．接続数 N_c が増えるほど誤報率と欠報率ともに減少し，所持ルール数が極端に少ない場合（$L_h = 50$）を除き，ホストは 10 台以上と相互テストすれば，適切な診断が可能であることが分かった．一方，所持するルール数 L_h が増えるほど欠報率はよくなるが，誤報率は変化しないことが確認された．誤報率が変化しない点は，接続数が少ないと正常な IDS が改ざん IDS に囲まれた状態になることがあり，たとえその正常な IDS が多くのルールを所持していても信用度が減少してしまうためだと考えられ，より詳細な理論的解析が必要である．

次に，モバイルエージェントの場合に対して，接続ホスト数 N_c の代わりに一ホストあたりのエージェント数 M/N，そして IDS が所持する平均ルール数 L_h を変化させて性能評価を行った．

Fig. 9 は，M/N と L_h を変化させた場合の 30 ステップ後の平均誤報率 $\bar{\alpha}$ と平均欠報率 $\bar{\beta}$（50 試行の平均）を

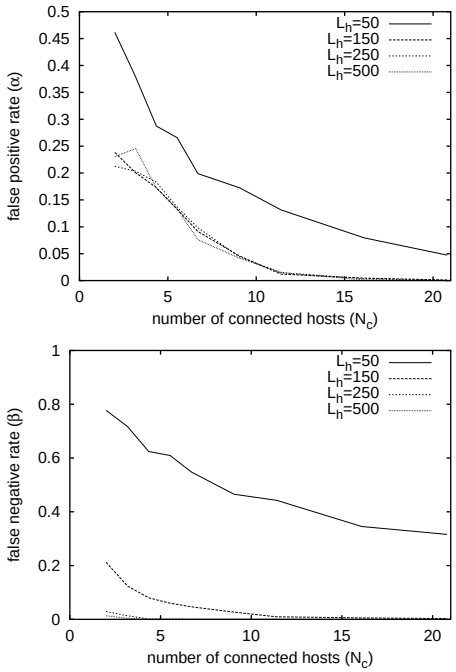

Fig. 8 Average false positive/negative rate ($\bar{\alpha}$ and $\bar{\beta}$) vs. number of connected hosts (N_c).

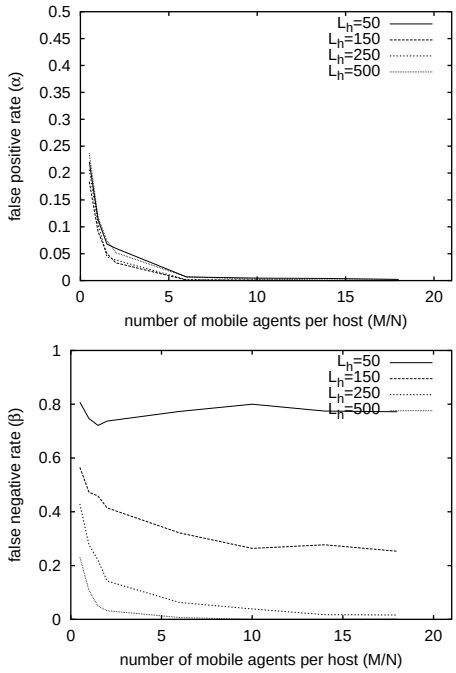

Fig. 9 Average false positive/negative rate ($\bar{\alpha}$ and $\bar{\beta}$) vs. number of mobile agents per host(M/N).

示す．所持ルール数が極端に少ない場合を除き，M/N が増えるほど誤報率と欠報率ともに減少し，一ホストあたり 5 個以上のエージェントがあれば，適切な診断が可能であることが分かった．ただし，所持ルール数が極端に少ない $L_h = 50$ の場合には，欠報率が減少しない．これは，ルール総数 5000 個に対して 50 個では同じルールを持つエージェントが存在し難く，$T_{ij} = 0$ となり信用度が変化し難いことに加えて，最初に改ざん IDS 上で生成されたエージェント（いわば自分の分身）がその改ざん IDS に戻ってきて，改ざんエージェントと改ざん IDS 間で $T_{ij} = 1$ が成立し，信用度が逆に増加しているためだと考えられる．しかし，所持ルール数 L_h が増えると，改ざん IDS に対して同じルールを持つ正常エージェントが増え，そのエージェントによって正しく診断されるといえる．

最後に，Fig. 8 と Fig. 9 を比較すると，ホスト間通信に比べてモバイルエージェントの方は誤報率は良いが，欠報率は悪い傾向にあるといえる．モバイルエージェントの誤報率が少ない点に関しては，診断は IDS ではなくエージェントによって行われるため，改ざん IDS に囲まれた正常な IDS も，正常なエージェントが移動してくることにより正しく診断されるためだと考えられる．所持ルール数が極端に少ない場合を除くとすれば，接続ホスト数（モバイルエージェント数）で見ると，より少ない数で誤報と欠報ともに小さいモバイルエージェントの方がより良いともいえる．

5 議論

本節では現時点での課題と今後の展開について議論する．

- 他のパラメータに対する検証
  Table 1 のパラメータのうち今回は固定したが，改ざん IDS の台数 N_f や改ざんの割合 e などに対する検証も必要である．
- 信用度の改ざん・偽装
  現時点では信用度や診断モデル自体の改ざんは考慮していない．例えば，Fig. 4 のやりとりにおいて，T_{ij} に関しては改ざん IDS はでたらめな結果を返すものの，信用度を偽装することまでは行っていない．実際改ざん IDS の信用度が正しく計算されるとは限らない．解決策の一つとして，一部集中処理となるが，各 IDS で求めたテスト結果 T_{ij} を最も信頼できる管理者マシンに送り，そこで信用度を計算することが考えられる．
- ルールの同一性の欠如
  本診断はルールの同一性に基づいているため，例えば Fig. 2 のラベル A に対して正しいデータが 1 だけでなく 2,3 などとある場合には，新たな仕組みが必要である．モバイルエージェントの場合では，エージェントの自律的生成と消滅により，同一性のあるルールを持ったエージェントだけが生き残り検査を続ける方法が考えられる．また，免疫型診断モデルをプラントのセンサネットに適用した例[15]では，センサ間の依存関係に基づいているため，ルールや IDS 間の関連性に基づく診断も考えられる．
- ルールの修正と改ざん
  今回はエージェントが持つルールは最初に固定され，移動中に正常 IDS による修正または改ざん IDS による改ざんはない．ルールの修正と改ざんを取り入れた場合のシミュレーションも必要である．

- 他の分散的診断手法との比較
  IDS の分野では分散的診断手法の議論は少ないものの，分散処理システムの故障を扱った基本問題として「ビザンチン合意問題」があり，いくつかの解決アルゴリズムが存在する[18),19)]．IDS の自己監視にこれらの手法も適用し，免疫型診断モデルと比較する必要がある．
- 実装レベルでの検証
  本シミュレーションシステムでは，現実を捉えきれていない事象が存在するため，現実の IDS やモバイルエージェントシステムを用いて検証する必要があり，現在実装中である．
- 移動のメリット
  今回はランダムな移動としたが，様々な移動戦略も用いてモバイルエージェントを使う場合と使わない場合を比較する必要がある．生物の免疫系で細胞が移動（循環）していることが，コンピュータネットワークのセキュリティに対しても有効であるかどうかである．Chess[20)]が指摘するように，モバイルエージェントそのものがセキュリティ問題となりうるため，それを補うだけの「移動のメリット」を示すことが重要である．

6 おわりに

本論文では，分散型 IDS において免疫型診断モデルを用いて自己監視を行い，ルールが改ざんされた IDS を検出した．シミュレーション結果から，ホスト間通信では，所持ルール数が極端に少ない場合を除き，ホストは 10 台以上と相互テストすれば，適切な診断が可能であることが分かった．一方，モバイルエージェントでは，正しいの診断のためには，所持ルール数が極端に少ない場合を除き，一ホストあたり 5 個以上のエージェントが必要であることが確認された．ホスト間通信とモバイルエージェントを比較すると，診断の収束速度ではホスト間通信の方が良く，診断の精度ではモバイルエージェントの方が良いことが分かった．今後は，**5** で列挙した課題に取り組む予定である．

謝辞

本研究は，国際コミュニケーション基金（調査研究助成）の支援を一部受けて行われた．

参考文献

1) 武田圭史, 磯崎宏: ネットワーク侵入検知，ソフトバンクパブリッシング, (2000).
2) 太田耕平, グレンマンスフィールド: インターネットにおける不正アクセス検出技術－ NIDS の現状と将来－, 信学論 (B), J83-B, 9, 1209–1216, (2000).
3) S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff: A sense of self for unix process, In *Proc. of 1996 IEEE Symposium on Security and Privacy*, 120–128, (1996).
4) S. Forrest, S. Hofmeyr, and A. Somayaji: Computer immunology, *Communications of the ACM*, 40, 10, 88–96, (1997).
5) S. Hofmeyr and S. Forrest: Architecture for an artificial immune system, *Evolutionary Computation Journal*, 7, 1, 45–68, (2000).
6) 溝口文雄, 大和田勇人, 西山裕之: 免疫系の超分散モデリングに基づく情報セキュリティ～その 2 不正侵入検出への応用～, 情報処理学会第 60 回全国大会, (2000).
7) M. Crosbie and E. Spafford: Defending a computer system using autonomous agents, In *Proc. of the 18th National Information Systems Security Conference*, (1995).
8) E. Spafford and D. Zamboni: Intrusion detection using autonomous agents, *Computer Networks*, 34, 547–570, (2000).
9) 浅香緑: モバイルエージェントによる侵入検出システムのための情報収集方式，信学論 (DI), J81-D-I, 5, 532–539, (1998).
10) G. Helmer, J. Wong, V. Honavar, and L. Miller: Intelligent agents for intrusion detection, In *Proc. of the IEEE Information Technology Conference*, 121–124, (1998).
11) D. Dasgupta: Immunity-based intrusion detection systems: a general framework, In *Proc. of the 22nd National Information Systems Security Conference*, (1999).
12) N. K. Jerne: The immune system, *Scientific American*, 229, 1, 52–60, (1973).
13) Y. Ishida: Fully distributed diagnosis by PDP learning algorithm: towards immune network PDP model, In *Proc. International Joint Conference on Neural Networks*, 777–782, (1990).
14) Y. Ishida: An immune network approach to sensor-based diagnosis by self-organization, In *Complex Systems*, Vol. 10, 73–90. Complex Systems Publication, (1996).
15) 石田好輝: 免疫型システムとその応用－免疫系に学んだ知能システム－, コロナ社, (1998).
16) 渡邊裕司, 石田好輝: 情報収集モバイルエージェントシステムにおける分散的信用度評価, 信学論 (DI), J85-D-I, 8, 758–766, (2002).
17) *Snort.org*, http://www.snort.org/.
18) L. Lamport, R. Shostak, and M. Pease: The byzantine generals problem, *ACM Trans. on Programming Languages and Systems*, 4, 3, 382–401, (1982).
19) 山下雅史: ビザンティン合意問題－信頼性の低い分散ネットワーク上での合意問題－, 情報処理, 32, 6, 682–693, (1991).
20) D. M. Chess: Security issues in mobile code systems, In G. Vigna, editor, *Mobile Agents and Security*, LNCS 1419, 1–14. Springer Verlag, (1998).