

ネットワーク侵入検出システム IDA の研究開発

Research and Development of IDA (Intrusion Detection Agent system)

浅香 緑*

女部田 武史†

井上 直‡

概要

本プロジェクトは、1999年4月から2002年3月まで、情報処理振興事業協会技術センターで、独創的先進的情報技術に関わる研究開発テーマのひとつとして実施された。侵入検出システム IDA (Intrusion Detection Agent system) の特徴は、軽量な侵入検出と侵入追跡機能にある。軽量な侵入検出のために IDA では、痕跡と判別分析を用いた検出手法を考案し、実装した。ここに痕跡とは侵入行為そのものではなく、多くの侵入行為に付随して発生するイベントである。本論文では IDA における侵入検出機能について述べる。

1 背景

侵入検出システム (IDS) は、認証・アクセス制御とならぶセキュリティ技術である。昨今はパケット情報から侵入を検知するネットワークベース型を中心に、普及しつつある。ネットワークベース型 IDS はパケット情報から、DoS 攻撃、ポートスキャン等の検出を中心に行う。一方のホストベース IDS は、ホスト上のシステムログから侵入を検出するため、システム内の詳細な情報から侵入が検出できる。またネットワーク内部からの侵入者の検出も可能である。IDS の研究が始まった 1980 年代当初は、ホストベース IDS の研究開発が主に行われていたが、1990 年代からはインターネットの普及に伴って、商用ではネットワークベース IDS が中心となっている。こういった状況のなかで、ネットワークベース IDS の欠点を補う視点から、ホストベースの IDS もまた見直されつつあるという状況である。

本プロジェクトではホストベース IDS である IDA (Intrusion Detection Agent system) の開発を行った。IDA では特権プロセス実行時に発生するシステムコールをおもに解析することにより侵入を検知する。これ

により軽量な侵入検出が可能になる。また IDA では侵入追跡機能も併せて持っている。本論文では、IDA の侵入検出機能についてのみ述べる。

2 目的

IDA [1, 2, 3] の研究開発では、侵入解析の負荷とログ取得の負荷を減らしつつ、できるだけ効率よく侵入検出を行うことを目標とした [4]。効率を向上させるために、システムの挙動全てを監視するのではなく、システムに重要な被害を与える攻撃 (管理者権限を奪われる等) を重点的に監視している。具体的には、まず痕跡の発生をモニタし、痕跡に関連した情報のみを解析することによって侵入を検出する。ここに痕跡とは侵入行為そのものではなく、侵入行為に付随して発生するイベントである。IDA では管理者権限が奪われる侵入を中心に検出を行う。そのために、痕跡としてはシステム管理者の権限に関連したものを対象とする。われわれの方法を用いるとシステムの挙動の全てを解析する必要がなくなるので、侵入解析の負荷を削減することができる。

われわれは痕跡を検出した後の情報解析の方法として、多変量解析の 1 分野である判別分析を用いた。判別分析を用いると、測定した対象を分類する関数が得られる。システムコールの発生状況を対象として判別分析することにより、通常行為と侵入行為の誤判別率を低く保ちながら分離することが可能である。さらに判別分析によって得られる分類関数を用いると、既知のリモートアタックだけでなく未知のリモートアタックも検出できる可能性がある。

3 痕跡

3.1 痕跡と侵入検出手法

われわれが開発した IDA は痕跡 (MLSI: Marks Left by Suspected Intruders) に基づく侵入検出技法を用いて実装されている。痕跡による侵入検出の基本的なアイデアは、攻撃ツールの収集、解析および、事例調査

*群馬大学工学部情報工学科 (376-8515 群馬県桐生市天神町 1-5-1, asaka@cs.gunma-u.ac.jp)

†株式会社日本総合研究所 (102-0082 東京都千代田区一番町 16, onabuta@sec.jri.co.jp)

‡株式会社 SRA 先端技術研究所 (160-0004 東京都新宿区四谷 3-12, now@sra.co.jp)

によって得られた [4]. 個々の侵入事例は多種多様なアプリケーションのバグ等を悪用しているが、侵入者が目的としている行為の種類は比較的少ない。たとえばパスワードファイルの入手や、管理者権限（特権）の入手である。つまり、全ての行為を網羅的に解析して侵入かどうかを判断するよりも、侵入者が目的としている行為、あるいは、侵入行為によって残された痕跡が検出されたときのみ侵入解析を開始する方が、効率的に侵入が検出できるのではないかと考えた [4]. 痕跡そのものは直ちに侵入を意味しない。たとえばパスワードファイルの変更を痕跡とすれば、正規の管理者によって痕跡が残る場合もある。このような場合には、痕跡を検出した後に情報を解析して、それが侵入かどうかを判断しなければならない。痕跡を検出した後の情報解析には、いろいろな侵入検出手法が応用できる。例えば痕跡の検出後に、既知の侵入パターンの知識から侵入かどうか判断するならば、それは MID に分類される。また通常行為からの逸脱から侵入と判断するならば、それは AID に分類される。痕跡による侵入検出はどちらの手法とも併せて使用することが可能である。本論文で提案する手法は、痕跡の検出後に、既知の侵入事例の知識によらず、統計的に侵入行為の群と通常行為の群を分離する。つまり AID に分類される手法である。

3.2 痕跡の定義

一言で言うならば、痕跡は侵入者が目標とする行為であり、侵入行為の結果発生するものである。痕跡は侵入そのものではないが、検出対象のほとんど全ての侵入で痕跡が残される。またできるだけ通常の事例では発生しないように定義されることが望ましい。

リモートアタック検出では痕跡の定義を、下記のように定めた。いずれもネットワークデーモンから起動される行為である。

- /bin/sh 等の起動（バッファオーバーフロー検出のため）
- /bin/cat 等の起動（任意のコマンド実行検出のため）

バッファオーバーフローによって残される痕跡は、ほとんど /bin/sh の起動である。それ以外にも /etc/shells に登録されているシェルについては、その起動をモニタする。なお任意のコマンド実行を検出するためには、厳密に考えると任意のコマンド全ての実行をモニタしなければならない。これは煩雑である。われわれはコマンドの中で特に cat 等パスワードフ

イルを閲覧可能なコマンド、および make 等システムに重要な影響を与える可能性のあるコマンドのみをモニタする。痕跡の検出およびその後の侵入検出のためにシステムコールのログが必要である。我々は Redhat Linux (kernel 2.2.14) 上に全てのシステムコールが取得可能なログツールを開発した。

本論文で解析に使用したデータは、侵入行為については、インターネット上から取得した攻撃ツールを用いた。また通常行為については、ネットワークデーモンが要求を受け付けた後に提供する一つの機能を単位として、これを無作為に 65 件選択して実行し、ログを取った。その中からほとんど全てのシステムコールを取得した。判別分析に使用したデータ（サンプル）はそのうち、痕跡を発生する侵入行為 12 件、通常行為 11 件である。われわれが取得している攻撃ツールでは全て上記のいずれかの痕跡が発生している。通常行為 65 件中 11 件で痕跡が発生した。以下に攻撃ツールとそれが残す痕跡を列挙する（表 1）。

表 1: 攻撃ツールと痕跡

ツール	侵入のタイプ	痕跡
named	Buffer Overflow	/bin/sh
amd	Buffer Overflow	/bin/sh
delegate	Buffer Overflow	/bin/sh
httpd	Buffer Overflow	/bin/sh
ipop2d	Buffer Overflow	/bin/sh
wu-ftpd	Buffer Overflow	/bin/sh
imap	Buffer Overflow	/bin/sh
pop3	Buffer Overflow	/bin/sh
pop3-list	Buffer Overflow	/bin/sh
sample CGI	execute arbitrary commands	/bin/cat
phf	execute arbitraty commands	/bin/cat
Count CGI	Buffer Overflow	/bin/sh

3.3 ログの切り出しの戦略

システムログには、システム内で同時に実行されているプロセスに関する記録が全て残る。各プロセスはプロセス ID を用いて識別できるが、異なるプロセス同士の関係（親子関係にあるのか、全く関係ないネットワーク接続のプロセスかなど）は、PID だけではわからない。また、一つの行為が一つのプロセスから構成されるとは限らない。一つの行為がどこから始まりどこで終るのかを識別することは容易ではない。こ

のような理由で、攻撃ツールによって発生するシステムコールや、通常行為の一つの機能を実行することによって発生するシステムコールを、一つのサンプルとして自動的に切り出すことは難しい。またサンプルを取得するために複雑な機能を実装すると、軽量の侵入検出を目標とする IDA の設計ポリシーに反する。そこで、判別分析を用いた侵入検出システムを実装するためには、サンプルをどのようにログから切り出すかを検討しなければならない。われわれは、複雑な処理を必要とせずサンプルとして切り出す方法を、システムコールの数、時間、プロセスの数、特徴的なシステムコールという視点から検討した。

侵入行為および通常行為のデータのシステムログの詳細を調べ、これらの観点で切り分け方法を検討した。その結果、以下が得られた。

- システムコールの数
調査対象とした 77 件 (通常行為及び侵入行為を合わせた全ての行為) の行為で発生したシステムコールの数は、最小 6 個、最大 3410 個であった。
- 時間
調査対象とした 77 件の行為の約 8 割が 1 秒未満、2 割弱が 2 秒以上の実行時間を要していた。
- プロセスの数
調査対象とした 77 件の行為のうち、一つのアクティビティで発生するプロセスの数は 1 から 12 個であった。
- 特徴的なシステムコール
調査対象とした 77 件の行為で、出現したシステムコールの数は 70 種類であったが、どの行為にも必ず出現するシステムコールというものは存在しなかった。

以上を総合して考えると、時間、プロセスの数、特徴的なシステムコールの観点で一つの行為を区切ることは現実的でない。われわれは総合的に考慮した結果、本論文ではサンプルとして、痕跡を残したプロセスが発生した時点から痕跡を残すまでの全てのシステムコールを切り出すことにした。

4 判別分析をもちいた侵入解析

4.1 判別分析の概要

痕跡は通常の行為から残されることもある。痕跡を検出した後には、痕跡を残す原因が正規に認められた

ものか、侵入行為かを判定する必要がある。この判定に判別分析を用いる。判別分析は多変量解析における 1 手法である。

判別分析は与えられたデータを規則的に分類する関数を与えるが、さらに新しいデータをその分類関数によって分類することも可能である。われわれは判別分析を侵入行為と通常行為の分類に使い、さらに新しいデータの分類についても考察する。母集団の性質により、分類関数が一次式 (線形判別関数 linear discriminant function) になる場合と、2 次式 (マハラノビスの距離 Mahalanobis' Distance) になる場合がある。判別分析は一般に以下の手順で行われる。

1. 説明変数 (predictor variable) の選択

説明変数とは、サンプルを構成する要素であり、目的変数 (本研究の場合は侵入行為であるかどうか) に影響を与える、その原因となるデータのことである。説明変数は基本的に次の観点で選択する [7]。

- 目的変数と相関が高いもの
- 互いに相関が高い説明変数はいずれかを削除

重回帰分析、判別分析では、自動的に変数を選択する変数選択法も存在する [8]。

2. 多変量正規性の検定

与えられた 2 つの群を分離するだけが目的であるならば、各群の多変量正規性の前提は必要ない。分類関数を求めるために使用したサンプル以外の、新しいサンプルがどちらの群に含まれるかを予測したい場合には、多変量正規性の前提が必要となる。各群が多変量正規分布をしていない場合は説明変数の変数変換等をおこない、各群の分布を多変量正規に近づける必要がある。

3. 分散共分散行列の相等性の検定

判別する 2 つの群の分散共分散行列が等しいかどうかを調べる。2 つの分散共分散行列が等しい場合は線形判別関数、等しくない場合はマハラノビスの距離によって判別を行う。本研究では侵入行為と通常行為の分散共分散行列が等しくないため、マハラノビスの距離を用いて判別を行なった。

4. 線形判別関数、またはマハラノビスの距離による各グループの分離

分類関数による分離結果と実績群（実際の結果）を比較して、見かけの誤り率（APER: Apparent Error Rate）を求める。

5. 分類関数の評価

これは現実の誤り率（AER: Actual Error Rate）を計算することで評価する。十分なサンプル数から分類関数を求めた場合は、見かけの誤り率と現実の誤り率の推定値が一致するとみて問題ない。しかし十分なサンプル数が得られない場合は、さらにジャックナイフ法による評価を行って、現実の誤り率の推定値を求めることができる。次節で詳しく説明する。

4.2 判別分析による通常行為と侵入行為の分類

判別の対象となる群を、リモートアタックと、リモートアタックではない通常のネットワーク行為とし、これらの分離を試みた。この節では一つの例に沿って、侵入事例と通常事例の分類の詳細について述べる。なお本研究で行った多変量解析は、エスミ社の多変量解析ツール [9] を用いた。

4.2.1 説明変数の選択と多変量正規性

侵入検出に判別分析を用いるに当たっては、説明変数として各システムコールを選ぶのが自然である。つまり、execve や open といったシステムコールを何種類か選択し、それらがある行為の中で何回発生したか数える。このようなデータを一つのサンプルとして判別分析を行うのである。しかしそのようにして得られたサンプルの群（母集団）は多変量正規分布とならない。よってこれらをそのまま説明変数として判別分析することはできない。本論文では、これらのシステムコールから多変量正規分布するような新しい説明変数を作る。

まず侵入と比較的高い相関を持つシステムコールを 10 ～ 20 種類選択する。一つのサンプルは痕跡から遡って、ある長さで切り取ったシステムログの中から、選択されたシステムコールが何回発生しているかを数えたものである。たとえばあるサンプルでは、fork 1 回、chdir 1 回、open 11 回、... が発生したというように数える。このようにして通常行為、侵入行為のサンプルを構成し、これらを主成分分析する。主成分分析で得られた新しい成分も、まだ多変量正規分布とならないので、これらをさらに変数変換して説明変数とする。これにより各群（侵入行為と通常行為の群）の

多変量正規性が得られる。多変量正規分布とすることにより、新しいサンプルがどちらの群に含まれるか、予測可能になる。この主成分分析、変数変換、多変量正規性の検定の詳細は付録で述べる。

4.2.2 分散共分散行列の相等性と 2 群の分離

前節で得られた説明変数を用いて、各群に対して分散・共分散行列の相等性のチェックを行う（付録参照）。本研究では分散・共分散行列が等しくならなかった。そこでマハラノビスの距離によって、サンプルがどちらの群に属するか判定した。ある点（観測値） X とある群とのマハラノビスの距離 D^2 とは、

$$D^2 = (X - \mu)' \Sigma^{-1} (X - \mu)$$

である。ここで、 μ は群の平均値、 Σ はその群の分散共分散行列（ Σ^{-1} はその逆行列）である。 x' は x の転置行列を表す。マハラノビスの距離とは、分散共分散を考慮した、観測値から群の重心までの距離である。ある観測値（サンプル）が、侵入行為と通常行為のどちらであるかを判別するためには、そのサンプルから侵入行為、通常行為の各群に対するマハラノビスの距離をもとめ、距離の近い方の群にその行為を分類すればよい。

ここでは、痕跡から遡って一つのプロセスから 400 個のシステムコールを切り出したサンプルを用いて、判別分析によって侵入行為と通常行為を分類した結果を示す（表 2）。この表において、番号という列は各サンプルの番号で、1 から 11 までが通常行為、12 から 23 までが侵入行為である。なおわれわれが持っている侵入行為のデータでは、全てが痕跡を残している。通常行為は痕跡を残しているもののみとりあげた。なぜならわれわれの侵入検出手法では、痕跡を残している行為に対してのみ侵入かどうかの判定が行われ、それ以外に対しては行われなからである。実績群とは実際にそれらのサンプルがどちらの群に属するか示しており、1 が通常行為の群、2 が侵入行為の群である。距離 1 とは各サンプルからの通常行為の群へのマハラノビスの距離であり、距離 2 とは各サンプルから侵入行為の群へのマハラノビス距離である。各サンプルは、より近い方の群に属すると見なされる。推定群とは、マハラノビスの距離からサンプルがどちらに含まれるか推定した結果である。実績群と推定群が一致した場合には、判別が正しく行われたことになる。例えばサンプル 1 において距離 1 と距離 2 を比べると、距離 1 のほうが小さい。距離 1 は通常行為の群への距離を表しているため、サンプル 1 は通常行為の群に近い、す

なわち通常行為の群に分類される。実際のサンプル 1 は通常行為であるので、この分類結果は正しいことになる。

表 2: 400 個で切り出した場合

番号	実績	距離 1 (通常)	距離 2 (侵入)	推定
1	1	9.090909	11.966177	1
2	1	2.424242	3.699446	1
3	1	2.424242	3.699446	1
4	1	4.090909	711.433249	1
5	1	4.090909	711.433249	1
6	1	9.090909	79.816633	1
7	1	9.090909	936.938096	1
8	1	2.424242	213.740312	1
9	1	2.424242	213.740312	1
10	1	2.424242	3.699446	1
11	1	2.424242	213.740312	1
12	2	2.424242	3.699446	1
13	2	8493.855690	5.616792	2
14	2	11290.439820	5.297369	2
15	2	301.344247	4.354192	2
16	2	16847.853862	4.124506	2
17	2	14841.839830	6.20569	2
18	2	372.605043	1.336114	2
19	2	374.632338	3.030727	2
20	2	353.819936	1.276899	2
21	2	17574.279637	7.116205	2
22	2	3753.931777	3.529735	2
23	2	11781.079121	9.412325	2

このようにして、この分類の誤り率を計算することができる。これを“見かけ”の誤り率 (APER) という。表 2 からこの判別の見かけの誤り率を計算すると、以下ようになる (表 3)。ここで“切り出した長さ”とあるのは切り出したシステムコールの数で、誤り率 1 とは、通常行為を侵入行為と誤って判別する (*false positive*) 見かけの誤り率、誤り率 2 とは侵入行為を通常行為と判別する (*false negative*) 見かけの誤り率、全体の誤り率は、両者を合計した全体の見かけの誤り率である。

4.2.3 ジャックナイフ法による分類結果の評価

見かけの誤り率は計算するのが簡単であるが、その値はしばしば現実の誤り率より低くなってしまふ。そ

表 3: 見かけの誤り率

切り出した長さ	誤り率 1	誤り率 2	全体の誤り率
400	0/11 (0%)	1/12 (8.3%)	1/23 (4.3%)

れは特にサンプル数が少ないときに生じやすい。その原因は、分類関数を作成するために用いたサンプルを、その評価にも用いていることにある。この問題を避けるために、分類関数を作成するサンプルと、評価するためのサンプルを分けるという方法がある。そのためには多数のサンプルが必要である。得られたデータのほとんど全てを分類関数作成に使わないと、重要な情報が落ちてしまう可能性があるという問題がある [6].

これ以外の方法として、Lachenbruch の差し出し法 (holdout procedure) [10] がある。これはしばしばジャックナイフ法と呼ばれる。この方法では、ほとんど全て (正確には全てのサンプルから一つを除いたもの) のサンプルから分類関数を構成するが、あるサンプルに対して評価を行う際は、このサンプルを除いたサンプルから作成された分類関数を使うというものである。具体的には次のようにする。

1. 二つの群, π_1, π_2 において、各群のサンプル数をそれぞれ n_1, n_2 とする。群 π_1 から一つのサンプルを除く。そして残りの $n_1 - 1, n_2$ 個のサンプルから、分類関数を作成する。
2. ステップ 1 における、分類関数作成に使われなかったサンプルを、その分類関数により分類する。
3. ステップ 1 と 2 を、全ての π_1 のサンプルに対して行って、これらを分類する。 $n_{1M}^{(H)}$ をこの群における誤分類されたサンプルの個数とする。
4. ステップ 1 から 3 までを π_2 のサンプルに対して行う。 $n_{2M}^{(H)}$ を、この群における誤分類された個数とする。

このようにして誤分類される全体の割合 ($n_{1M}^{(H)} + n_{2M}^{(H)} / (n_1 + n_2)$) は適度のサンプルに対して、現実の誤り率の期待値のほとんど不偏な推定値になっている [6]。このジャックナイフ法を用いて 4.2.2 で分類した分類関数の評価を行う。その結果を表 4 に記す。

表 4 において、サンプルとは差し出された (除かれた) サンプルの番号を指す。差し出されたサンプルは分類関数を求めるためには使われない。差し出されたサンプル以外のデータから分類関数をもとめ、求められた分類関数により、そのサンプルが二つのうちのど

ちらに分類されているかを予測する。その結果が表の各行に記されている。例えばサンプル番号1の場合、サンプル番号1以外の22個のサンプル（通常行為10件、侵入行為12件）から求められた分類関数（マハラノビスの距離）をみると、侵入行為の群（距離2）より通常行為の群（距離1）に近い。よってサンプル番号1は通常行為と分類される。サンプル番号1のサンプルは、この分類関数生成には使われていないので、サンプル番号1を正常行為と判断することに関して、サンプル番号1の情報は全く使われていない。このようにして、全てのサンプルに対して分類が正しく行われているか否かを計算し、この分類関数の現実の誤り率を推定することができる。このケースではサンプル番号12のみ、本来侵入行為であるものを通常行為と分類した。よってこの分類の現実の誤り率の推定値は、4.3%となる。

表 4: ジャックナイフ法の結果

サンプル	群	距離 1 (通常)	距離 2 (侵入)	予測
1	1	5.801640	11.968080	1
2	1	3.600100	3.698625	1
3	1	3.600100	3.698625	1
4	1	8.105604	711.437887	1
5	1	8.105604	711.437887	1
6	1	0.393969	79.821200	1
7	1	8.902336	936.958068	1
8	1	3.599656	213.749684	1
9	1	3.599656	213.749684	1
10	1	3.600100	3.698625	1
11	1	3.599656	213.7497	1
12	2	2.424316	6.320504	1
13	2	8494.132281	13.718479	2
14	2	11291.012375	12.075152	2
15	2	301.332864	8.290551	2
16	2	16847.722663	7.550893	2
17	2	14841.452685	17.459302	2
18	2	372.665580	1.666848	2
19	2	374.607071	4.686863	2
20	2	353.841837	1.581724	2
21	2	17574.125953	26.163890	2
22	2	3753.667718	5.876572	2
23	2	11781.079121	153.023198	2

4.3 システムコールの切り出し数による分類結果

ここでは痕跡を残したプロセスに含まれるシステムコールの数を、痕跡から遡ってある数で切ることによって、侵入行為か通常行為かの分類結果にどのような影響があるのか分析してみた。行為によって、一つのプロセスに含まれるシステムコールの数はかなり異なる。サンプルを何らかの数で区切り、ある程度同じ数のシステムコールで各サンプルを比較することは、侵入検出精度の向上（誤り率の低下）につながると思われる。切り出しは痕跡を起点として、時間を遡る形で行った。試したパターンは以下の通りである。

- 痕跡を残したプロセスから、痕跡検出時から遡って、全てのシステムコールを切り出す
- 痕跡を残したプロセスから、痕跡検出時から遡って、それぞれ、1000個、800個、700個、600個、500個、400個、300個、200個、100個のシステムコールを切り出す

ある行為に含まれるシステムコールの数が切り出す数より短いときは、短いほうをサンプルとすることとする。たとえばシステムコールの総数を100個で切っている場合に、あるサンプルではシステムコールが50個出現しなかったとする。この場合はそのサンプルは、50個のシステムコールから構成されるとする。判別分析による分類方法は、前節で述べたものと同様である。これらに対して、見かけの誤り率およびおよび現実の誤り率を求めた（表5）。表中で全てとあるのは、一つのプロセスで発生した全てのシステムコールを切り出した場合である。これを見ると多少のばらつきがあるものの、400個前後の誤り率が低く、それより多いときも少ないときも誤り率が高くなるのがわかる。特に現実の誤り率においてこの傾向が顕著に現れている。このことから、痕跡の近辺のある範囲に、通常行為と侵入行為の違いが大きく現れていると考えることができる。範囲が広すぎた場合は分類に不必要な情報が増え、逆に少なすぎる場合は必要な情報が落ちてしまうものと思われる。

5 考察

本論文は、不正侵入を検出するために、まず痕跡を検出し、その痕跡を残した行為が侵入であるかを判別分析を用いて判定する手法を提案している。この手法には以下の特徴がある。それぞれの特徴ごとに述べる。

表 5: 全体の誤り率

切出数	見かけの誤り率	現実の誤り率
100	21.74 %	13.04 %
200	21.74 %	21.74 %
300	4.35 %	17.39 %
400	4.35 %	4.35 %
500	21.74 %	17.39 %
600	17.39 %	21.74 %
700	17.39 %	21.74 %
800	21.74 %	26.09 %
1000	26.09 %	30.43 %
全て	17.39 %	39.13 %

5.1 判別分析の特徴

- 判定ルールのアップデートの必要がない

MID による検出方法では、未知の侵入方法が発見されるたびに判定ルールを追加しなければならない。また AID でも一部の手法では、対象とするシステムのアプリケーション毎に判定のためのデータベースを作成しなければならない。この場合には、システムに新しいアプリケーションを導入する毎に、データベースを更新し、あるいは追加をしなければならない。本論文の手法では、ネットワークデーモン全般の行為について統計的な手法により侵入判定を行っている。よって新しい侵入手法の発見や新しいアプリケーションの導入に伴って、判定ルールをアップデートする必要がない。

- データベースやプロファイルの必要がない

一般的に AID ではプロファイルが必要である。これらの作成のためにシステムの開発が必要であったり、更新したり維持するためにもコストがかかる。本手法では、侵入行為かどうかの判定はターゲットのサンプルが通常行為の群と侵入行為の群のどちらに近いマハラノビスの距離によって求められる。よって判定のために、データベースやプロファイルは必要ない。

- 未知のリモートアタック検出が可能である

判別分析では母集団が多変量正規であるという前提を満たすとき、新たなサンプルに対して予測が可能になる。すなわちわれわれの手法では、未知のリモートアタックが検出可能である。つまり一度侵入と通常行為の群を分離した後、新しいデー

タ (サンプル) から各群へのマハラノビスの距離を求めれば、それが侵入か通常かどうかを判定できる。

- システムコールの発生順序に関係ない手法である
本手法では、システムコールのログから侵入検出を行う。システムコールのログから侵入検出を行うシステムは多数存在するが、それらはいずれもシステムコールの発生順序を、何らかの形で侵入判定に用いている。本手法では、ある長さのシステムログの中にどの種類のシステムコールがいくつ発生したかという情報が必要であって、判定のために発生順序は必要ない。本手法では、判定に必要なデータの取得および判定が容易である。
- 判定機能を迂回することが困難である

攻撃者が意図的にこの手法を迂回するのは困難である。本手法では 10 数種類のシステムコールの情報から侵入行為か通常行為かを判定している。これらのシステムコールの発生数を、通常グループに属するように保った上で、侵入行為を実現することは極めて困難である。例えば、これらのシステムコールを全く使わない攻撃を考えても判定を迂回できない。なぜならば判別関数の中には、通常行為かどうかの指標となる項も存在するからである。あるシステムコールを全く使わなければ、それは一つの特徴となってしまふ。

なお本論文では判別分析のために 10 数種類のシステムコールを選択した。その種類や数がこれと全く同じでなくても、判別分析で似たような結果を出すことが可能である。すなわち各サイト毎に異なるシステムコールを選択して侵入検出を行うようにすれば、迂回はさらに困難となる。

5.2 痕跡による特徴

- サンプル切り出しの容易性と検出精度

本文中でも指摘した通り、システムログ中から機械的に“一つの行為”を切り出すことは、容易ではない。痕跡を起点として一つのプロセスから決まった数のシステムコールを切り出すことにより、侵入判定のためのサンプルを容易に切り出せる。また痕跡を起点として、400 個程度のシステムコールを切り出した場合は、個々の行為で発生するシステムコール全てをサンプルとするよりも、侵入精度が良くなる。

- 痕跡による解析対象の絞り込み

本手法では痕跡を発生させたプロセスのみ，侵入検出の解析対象とする．すなわち，全てのネットワークアクティビティを解析せずに，限定された行為（痕跡を残す行為）のみ解析することが可能になる．これにより，侵入行為と全ての通常行為を分類するより，分類精度が向上することが考えられる．また検出ターゲットを絞ることにより，フォルスポジティブの影響を少なくすることが可能になる．つまり仮にフォルスポジティブの比率が10%とする場合，1000件を解析すれば誤判断が100件生じるが，痕跡により解析対象がこの1000件のうち100件に限定されていれば，誤判断は10件に減る．

• 痕跡の発生率

われわれの実験環境で，ある時間帯に発生したイベント数は22483回であった．そのうち痕跡は4回発生した（0.018%）．全ての痕跡から400個のシステムコールを切り取った場合でも，全データの1600/22483（7.1%）の解析をすれば良い．この数値は環境に依存するものであるが，提案手法が効率的な検出法である一つの例となる．

• 痕跡近辺の異常性

Forrestらが指摘しているように[11]，侵入行為における，システムコールの通常性からの逸脱は，侵入行為全般においてまんべんなく発生するのではなく，ある箇所に集中的に発生する．本論文の研究によっても，痕跡の近辺に集中して異常性が検出されることが分った．

6 おわりに

本論文では，IDAにおけるリモートアタック検出機能について述べた．痕跡の定義を行い，痕跡検出後に判別分析によって侵入を判定する方法について述べた．一定の数のシステムコールを切り出すことにより，侵入精度が向上することがわかった．また痕跡の検出後に，侵入検出を開始することにより，解析の負荷を減らすことができた．

IDAではこの他，侵入追跡機能，必須アクセス制御を用いた防御機能を持っている．これらの詳細については[12, 13]を参照されたい．

参考文献

[1] <http://www.ipa.go.jp/STC/ida>

- [2] 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 (D-I), Vol.J81-D-1, No.5, pp.532 - 539, May 1998.
- [3] M. Asaka, "Information-Gathering with Mobile Agents for an Intrusion Detection System," Systems and Computers in Japan, Vol. 30 No. 2, pp.31 - pp.37, Feb. 1999.
- [4] M. Asaka, M. Tsuchiya, T. Onabuta, S. Okazawa, and S. Goto, "Local Attack Detection and Intrusion Route Tracing," IEICE Trans. Commun., Vol. E82-B No.11, pp.1826-1833, Nov. 1999.
- [5] M. Asaka, T. Onabuta, T. Inoue, S. Okazawa, and S. Goto, "A new intrusion detection method based on discriminant analysis," IEICE Trans. Inf & Syst., Vol. E84-D No.5, pp.570 - 577, May 2001.
- [6] R. A. Johnson and D. W. Wichern, "Applied multivariate statistical analysis," Fourth Edition, Prentice Hall, New Jersey, 1998.
- [7] 菅 民郎, "多変量解析の実践," 現代数学社, 京都, 1993.
- [8] 大澤 清二, 稲垣 敦, 菊田 文夫, "生活科学のための多変量解析," 家政教育社, 東京, 1992.
- [9] <http://www.esumi.co.jp>
- [10] P. A. Lachenbruch and M. R. Mickey. "Estimation of Error Rate in Discriminant Analysis." Technometrics, 10, no.1 pp.1-10, 1968.
- [11] S. A. Hofmeyr, A. Somayaji, and S. Forrest, "Intrusion detection using sequences of system calls," Computer Security Vol. 6, pp 151-180, 1998.
- [12] M. Asaka, T. Onabuta, and S. Goto. "Public Information Server for Tracing Intruders in the Internet," IEICE Trans. on Commun., Vol. E84-B No.12, pp.3104 - 3112, Dec. 2001.
- [13] 女部田 武史, 井上 直, 浅香 緑, "必須アクセス制御方式を用いた侵入検出システム保護機能," 情処学論, Aug. 2001.