卒業論文　　2010年度（平成22年度）

IDSログの統計的解析を利用した異常検知システム

慶應義塾大学 環境情報学部

氏名：Doan Viet Tung

担当教員

慶應義塾大学 環境情報学部

村井 純

徳田 英幸

楠本 博之

中村 修

高汐 一紀

重近 範行

Rodney D. Van Meter III

植原 啓介

三次 仁

中澤 仁

武田 圭史

平成23 年2 月14 日

卒業論文要旨 - 2010 年度 (平成 22 年度)

IDS ログの統計的解析を利用した異常検知システム

インターネットが通信のインフラとして普及したことにより，ネットワークで重要な情報のやりとりが行われるようになった．しかし，その反面，情報化社会において，システムの脆弱性を利用したコンピュータやネットワークへの不正アクセス，コンピュータウィルス感染などといった情報セキュリティに関する問題が急増している．不正アクセスやコンピュータウィルスによって，システムの停止や内部情報の流出などの危険があることは周知の通りである．そこで，このようなネットワークセキュリティ確保のために侵入検知システムなどの対策が重要視されつつある．

侵入検知システム（IDS）とは，ネットワークを流れるパケットを監視して，不正アクセスと思われるパケットを発見したときにアラームを表示するシステムである．しかし，既存の IDS では，正しい通信を不正な通信と判断する誤検知（フォルスポジティブ），不審な通信を正しい通信と判断する誤検知（フォルスネガティブ）を含んだ大量のログを生成するため，これらをすべて確認することは管理者には大きな負担となる．

本論文では，侵入検知システムが異常な検知数を検出するときに，過去の警告イベントの傾向に基づいた，ログ分析手法を提案する．具体的には，時間帯に統計し，警告イベントごとに過去の検知数の平均と標準偏差を計算する．これによって，検知数が異常かどうか判断することができる．警告イベントごとに異常レベルを算出し，管理者に注意すべき警告イベントと無視すべき警告イベントを分類する．これによって，管理者が IDS を運用する際に，管理コストを低減することができる．そして，提示した手法を検討するために，異常な警告イベントを検出し，検知数がどう減るかを評価した．

キーワード:
1. セキュリティ, 2. 侵入検知システム, 3. 異常検知, 4. 統計分析,

慶應義塾大学 環境情報学部

ドアン ヴィエット トゥン

Abstract of Bachelor's Thesis - Academic Year 2010

Anomaly Detection System using Statistical Analysis of IDS logs

Nowadays, with the spread of computer networks, information systems are more open to the Internet. Therefore, the importance of secured networks is tremendously increased. There has been an increasing need for security systems against the external attacks from the hackers. One important type is the Intrusion Detection System (IDS).

Intrusion Detection System tries to detect malicious activities such as denial of service attacks or even attempts to crack into computers by monitoring network traffic. IDS reads all the incoming packets and tries to find suspicious patterns known as signatures or rules. However, IDS generally generates many false positives (corresponding to a false alert) and false negatives (corresponding to a non-detected attack). As a result, it's a heavy workload for the network manager to check all the detected events.

This thesis presents an anomaly detection system using statistical analysis of IDS logs. Based on tendency of IDS alert events, this system can cluster, analyze and then classify alerts into normal alerts and abnormal alerts.

Keywords :
1. Security, 2.Intrusion Detection System, 3.Anomaly Detection, 4.Statistical analysis

Keio University, Faculty of Environment and Information Studies

Doan Viet Tung

目 次

図 目 次

表 目 次

第1章　序論

1.1　ネットワークセキュリティの現状

現在，インターネットのブロードバンド化によって，通信のインフラが普及した．これにより，ネットワークへの不正アクセスなどの情報セキュリティに関する問題は急増し，ネットワークセキュリティの重要性が認知されてきている．

独立行政法人情報処理推進機構（IPA）[1] によると，2010 年の年間届出件数は 197 件となり，2009 年の届出件数 149 件から 48 件（約 32 %）増加した．表 1.1 は，2 年間の不正アクセス届出件数とウィルス届出件数である．

		2009 年	2010 年
不正アクセス	届出	149 件	197 件
	相談受付	420 件	601 件
	合計	569 件	798 件
ウィルス届出		16,392 件	13,912 件

表 1.1: 不正アクセス・ウィルス届出件数

コンピュータウィルスの届出件数に比べて，不正アクセスの届出件数が少ない．ただし，不正アクセスが少ないからではなく，直ちに目に見えるような異常が現れたわけではなく，他のシステムに対する攻撃を踏み台にされることやデータを盗聴されることなどの不正なアクセスもあるため，被害者が気づいていない時，その不正アクセスを届出できない．つまり，不正アクセスの件数は表 1.1 での数より多いと思われる．

表 1.2 から，2010 年で不正アクセスの被害あったケースは合計で約 28 %増加したことが分かる．その中でも侵入は，2010 年（67 件）と 2009 年（36 件）を比べると，大幅に増加した（約 86 %）．

代表的な不正アクセスの手法として，侵入，アドレス詐称，DOS 攻撃 [2] などがあり，被害の内容としては，プログラム埋め込み及びなりすましが多かった．2010 年の不正アクセス届出件数の中に，侵入を 54%，なりすましを 28%占めている．ウェブページ内に，悪意あるサイトへ誘導するためのスクリプトが埋め込まれているのが行われた．改ざんされたのは，ファイル名に index という文字が含まれるもの（index.html や index.php な

届出種別	2009 年	2010 年
侵入	36 件	67 件
なりすまし	32 件	35 件
DOS 攻撃	5 件	7 件
不正プログラム埋め込み	12 件	6 件
アドレス詐称	2 件	3 件
その他	9 件	5 件
合計（被害あり）	96 件	123 件

表 1.2: 不正アクセスの届出種別

ど）や，JavaScript[3] 外部ファイル（例：xxx.js）であった．なりすましの被害は，オンラインサービスのサイトに本人になりすまして何者かにログインされ，サービスを勝手に利用されていたものを発見した．

被害原因について，2010 年で実際に被害があった届出を原因別分類（表 1.3）により，ID・パスワード管理・設定の不備が 16 件（13 %），古いバージョン使用・パッチ未導入などが 13 件（10 %），設定不備が 7 件（6 %），となっています．原因が不明なケースは 75 件（61 %）と全体の半数を超えており，また，2009 年と比べて 21 件（約 39 %）も増加していることから，2010 年においても，不正アクセスの手口の巧妙化および原因究明が困難な事例が多くなっているということが推測される．

被害原因	2009 年	2010 年
ID，パスワード管理の不備	11 件	16 件
古いバージョン使用，パッチ未導入など	16 件	13 件
設定不備	6 件	7 件
不明	54 件	75 件
その他（DoS など）	9 件	12 件
合計（件）	96 件	123 件

表 1.3: 被害原因

一般的に，ネットワークを流れるパケットを監視して，不正アクセスを発見するために，侵入検知システム（Intrusion Detection System - IDS と呼ぶ）を利用する．ただし，不正侵入を検知すると，管理者に通報できる侵入検知システムは誤検知を含んだ膨大なロ

グを生成すると言われている．誤検知では，正しい通信を不正な通信と判断するフォルスポジティブと不正な通信を正しい通信と判断するフォールスネガティブがあり，これらを調整せずに，管理者は大量の警告をすべて確認しなければならず大きな負担となる．

1.2 本研究の目的

本論文の目的は，管理者の負担が少なくなるように侵入検知システムのログを調整する手法を提示することである．どのような警告イベントに注意するべきであるか，どのような警告イベントを無視しても問題ないか，ログ分析方法を言及する．提案手法を実験に適用し，注意すべき警告イベントと無視してよい警告イベントを分類する手法を検討する．

毎日，侵入検知システムから膨大な警告イベントを管理者に通知する．その中に，短い時間にも複数に検知された警告が多数あるため，一定の期間の警告イベントをクラスタリングすることで，警告数をある程度削減することを考える．それに，警告イベントの過去の傾向を分析することにしたがって，警告の検知数は異常なものかどうか，判断する可能がある．本論文では，データマイニング技術を用いて，シグネチャごとに過去の検知数の分布を調査し，その結果に基づいて異常なものと正常なものに警告イベントを分類する手法を挙げる．

そのうえで，正常な警告イベントを無視することと異常な警告イベントを注意を払うことのメリットは何か検証・考察する．しきい値を設定することによって，一部分の正常な警告を除去することで，侵入検知システムの誤検知の割合がどう変化するか説明する．

1.3 本論文の構成

本論文は全 6 章から構成される．第 2 章では，侵入検知システムの基本的な性質，検知対象による分類，検知アルゴリズムによる分類および誤検知の問題について述べる．第 3 章では，第 2 章で述べた課題に取り組む関連研究を紹介する．第 4 章では，管理者の負担を減らすために警告イベントの過去の傾向に基づく侵入検知システムのログ分析手法を提案する．第 5 章では，第 4 章で述べた手法を実際に適用し，実験結果を述べるとともに，評価・考察を行う．最後に第 6 章で本論文の結論と，今後の展望を述べる．

第2章　既存技術とその問題点

本章では，ネットワーク上のトラヒックを監視するIDSの基本的な性質，監視する対象による分類，分析方法による分類について述べた．そして，IDSの誤検知と管理者の負担の関係について述べた．最後に本論文の着眼することを示した

2.1　Intrusion Detection System

IDSとはIntrusion Detection Systemの略で，日本語では侵入検知システムと呼ばれている．IDSはネットワーク上などへの不正なアクセスの兆候を検知し，ネットワーク管理者に通報する機能を持つソフトウェア，またはハードウェアである．管理者は実際の被害に先だって警戒でき，必要なら回線切断等の防衛策を講じ，システムの破壊などを未然に防止できる．IDSが自動的な通信切断やサーバのシャットダウンなどの防衛を実行せず，あくまで管理者にメールを送信して異常を通知するだけなのは，検知した異常が攻撃であるとは限らないためである．最も大きなIDSを導入するメリットは異常に気付かせてくれるという点であり，もしIDSがなければ，何か異常が発生したとしてもそれに気付くことは容易ではない．

しかし，すべての警告が不正なアクセスであるとは限らず，その中に誤検知の膨大なログが生成される．一般に，誤検知といわれるものには，大きく分けて2つの種類が存在する．フォールスポジティブとフォールスネガティブである．IDSを最適化するために，誤検知数を減らさなければならない．

実際のIDSは監視対象により，以下の2つに分類される．

- ホスト型（HIDS）
  ホストに到達したパケットをそのホストが検証する．
- ネットワーク型（NIDS）
  ネットワーク上のすべてのパケットを監視し，不正なパケットを検出する．

HIDSとNIDSの構成を図2.1に示す．

一般的なNIDSは主に，不正検知と異常検出という二つの手法を用いて攻撃や不正アクセスを検知する．不正検知（別名：シグネチャ・ルールベース検出）とは，取り込んだパケットとNIDSに登録された複数のシグネチャとを比較し，不正アクセスを検知する手法（Misuse Detection）である．または，異常検出（Anomaly Detection）とは，取り込んだパケットをRFCのプロトコル仕様などを比較し，仕様から逸脱したものを異常として検知する手法である．

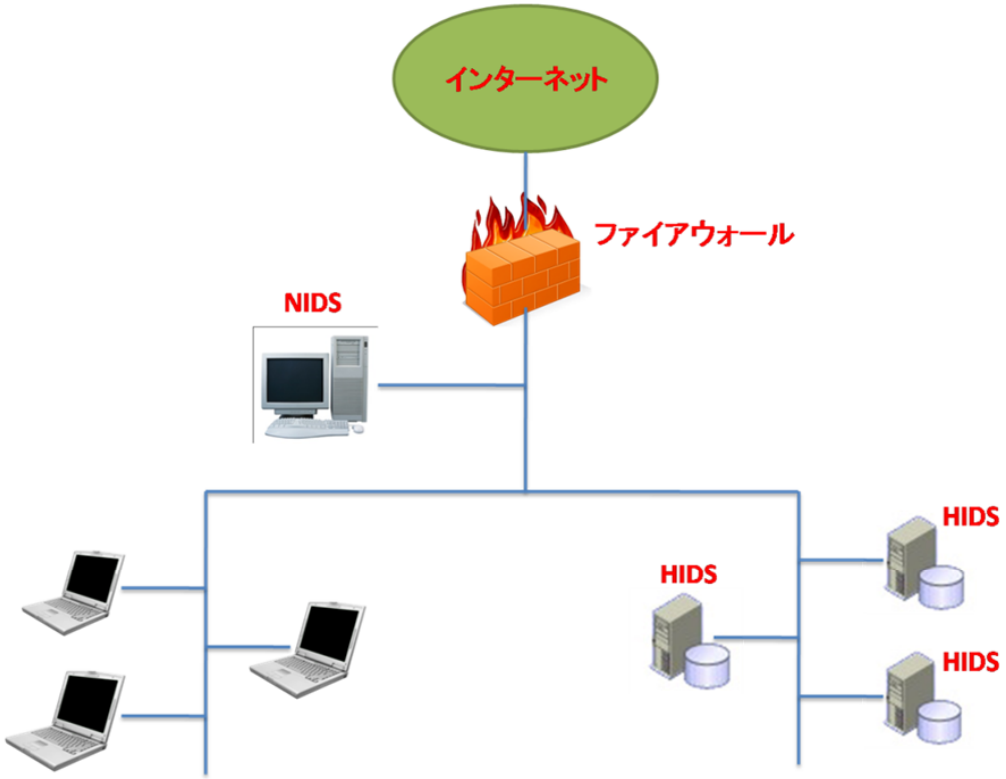


図 2.1: NIDS & HIDS

カリフォルニア大学デービス校による共通侵入検知フレームワーク（CIDF：Common Intrusion Detection Framework[4]）では，4 種類の機械からなるコンポーネントを用いた侵入検知システムの構成を提案している．このフレームに基づく一般的な侵入検知システムのモデル図を図 2.2 に示す．

図 2.2 のような侵入検知システムでは，各構成要素は以下の 4 つに分類できる．[5]

- イベント生成部：Event generator（E-box）
  システム環境の中から，侵入検出に必要なイベント情報を入力として獲得する．
- イベント分析部：Event analyzer（A-box）
  侵入を検出するためにイベント解析を行うモジュール．さまざまなタイプの分析手法をこのモジュールに適用する．
- イベントデータベース：Event database（D-box）取得したイベント情報を格納しておくデータベース
- レスポンスユニット：Response unit（R-box）検出結果に基づいて，プロセスの停止，接続の判断，ファイル設定変更，管理者への通知などの侵入に対する対処を実施するためのモジュール．

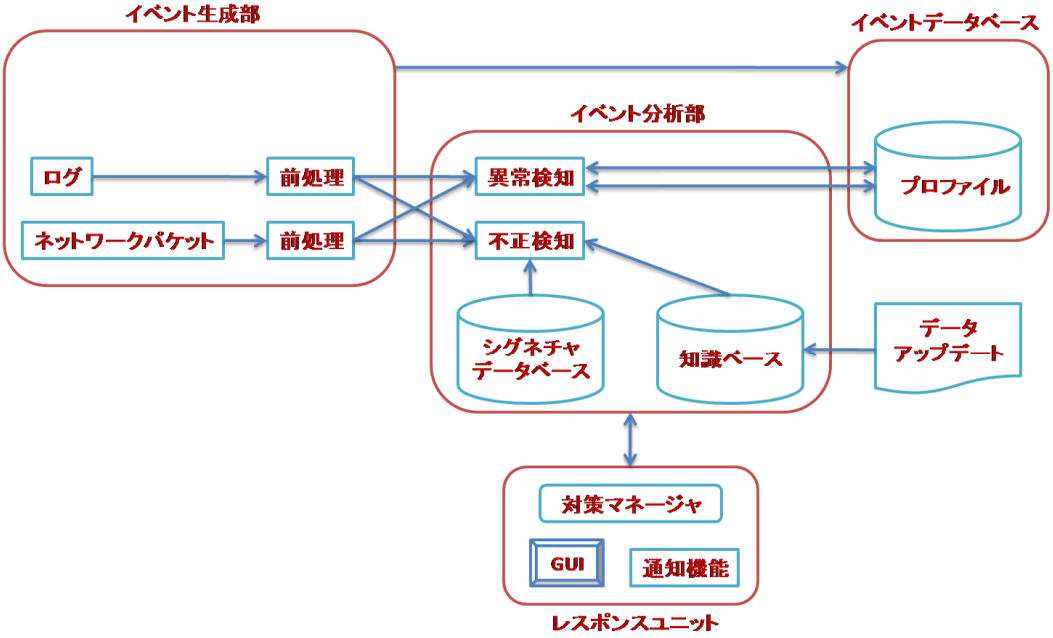


図 2.2: 侵入検知システムの構成

2.1.1 誤検知

IDS の誤検知の割合を測るための指標として，フォールスポジティブ（false positive）とフォールスネガティブ（false negative）の二つがある．フォールスポジティブとは，IDS が本来検知すべきではない，つまり不正ではない事象を不正行為として検知してしまうことを指す．フォールスネガティブとは，本来検知すべき不正行為を見逃してしまうことを指す．

IDS が警告をログに記録する場合，誤検知の相対量が次第に増えてくると，正規の警告が誤検知に埋もれてしまう．それで，誤検知を無視せずに，極力減らすべきである．しかし，フォールスポジティブを低く抑えようとすると，フォールスネガティブの発生確率が増加し，逆にフォールスネガティブを低く抑えようとするとフォールスポジティブの発生確率が増加して運用管理者への負担が大きくなる傾向がある．そのことによって，IDS の運用においては，フォールスポジティブとフォールスネガティブを最小にするよう継続的にチューニングを提案する必要がある．

警告イベントと不正アクセスと誤検知の関係を図 2.3 に示す．

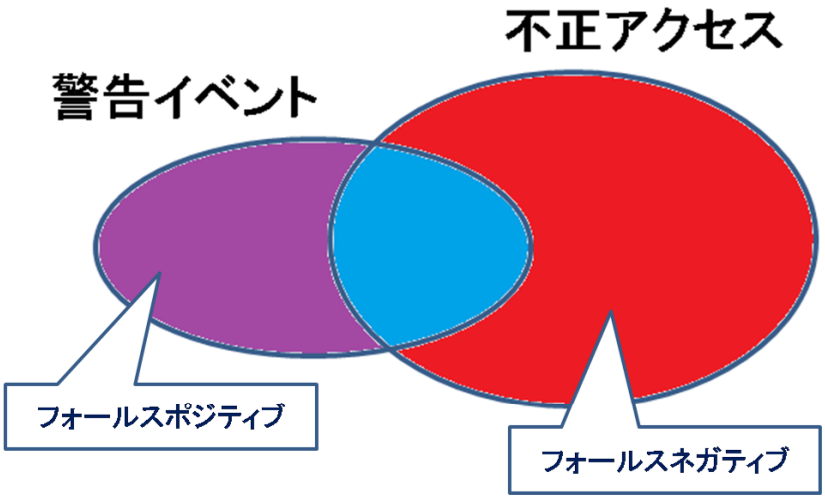


図 2.3: 警告イベント，不正アクセスと誤検知の関係

2.1.2 検知対象空間による分類

IDS は，その動作形能や監視対象によって，ネットワーク型侵入検知システムとホスト型侵入検知システムの二つに分類することができる．

1. ホスト型侵入検知システム（Host-based Intrusion Detection System - HIDS）
   ホストベース侵入検知システム（HIDS）は，単独のホスト上でプロセスやログ（システムログやアプリケーションログやモジュールが生成するログなど）を監視し，各種のログに不審な振る舞いが観測された場合に警告が発する．つまり，HIDS は自らが動作しているホストのみの保護を行い，攻撃や侵入によって引き起こされた結果を常に監視するシステムである．

   このほかに，ネットワーク上に分散した複数のホストベースシステムを連携させ，情報交換などを行うものをマルチホストベースシステムと呼ぶばあいがある．また，OS 上でアプリケーションが使用するシステムコールの情報を取得し，その情報を入力とするものをプロセスベースシステムと呼び，ホストベースシステムから派生した形態と位置づけることができる．

2. ネットワーク型侵入検知システム（Network-based Intrusion Detection System - NIDS）
   ネットワークベース侵入検知システム（NIDS）は，監視専用の機器を監視対象となるネットワークセグメントに接続して使用する．その機器は，自身が接続されたネットワークを流れるパケットをリアルタイムに監視し，あらかじめ設定されたルールに基づいて不正なアクセスや不審な事象などを検知する．検知した結果は管理用の

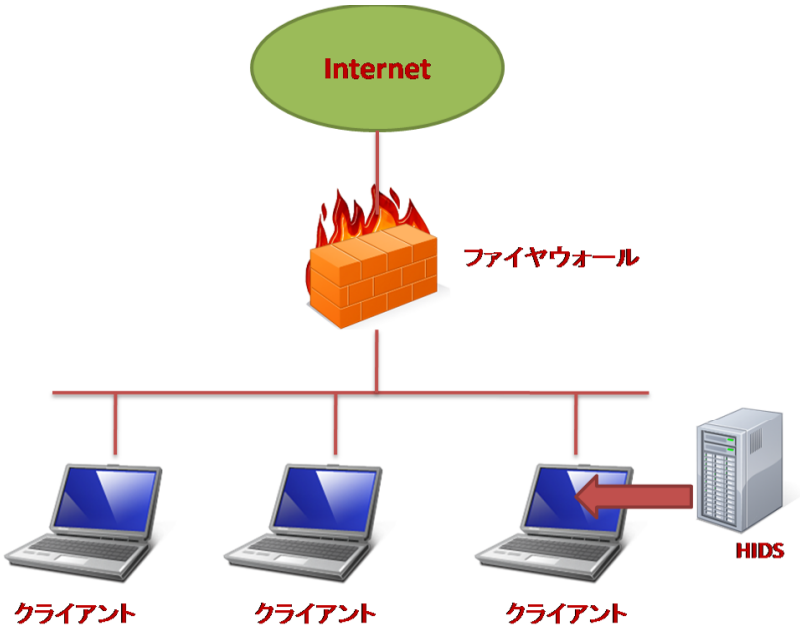


図 2.4: HIDS の構成図

ソフトウェアがインストールされた端末の画面に通知したり，メールによって通知したりすることが可能である．

NIDS には，自ホスト宛のパケットのみを監視するもの，同一ネットワーク内の他ホスト宛のものを含む全トラフックを監視するものの 2 つのタイプがある．また，パケットのヘッダー情報のみを見て検出処理を行うものと，トラフィックの内容までを監視対象とするものがある．ネットワークの全トラフィックを監視する場合，ネットワーク内での配置や取得データの選択が検出に大きく影響することになる．

NIDS の機能上の限界や運用上の課題を以下に多少示す．

- 暗号化されたパケットは解析できない
  通信パケットが暗号化されたとき，NIDS が理解できない．したがって，HTTPS 通信 [6] における攻撃や不正なアクセスを検知するには，SSL アクセラレータ [7] を導入する必要がある．その他の暗号化通信については，個別に検討が必要である．
- 攻撃は検知できても侵入を検知できない
  NIDS は攻撃が行われたことが検知するが，実際に侵入が行われたことが検知できないことが多い．
- 不正なアクセスを防御できない
  インライン接続で使用する機器ではないため，NIDS で攻撃を排除するには限界がある．不正アクセスを防御するために，IPS[8] やウェブアプリケーションファイアウォール [9] を利用しべきである．

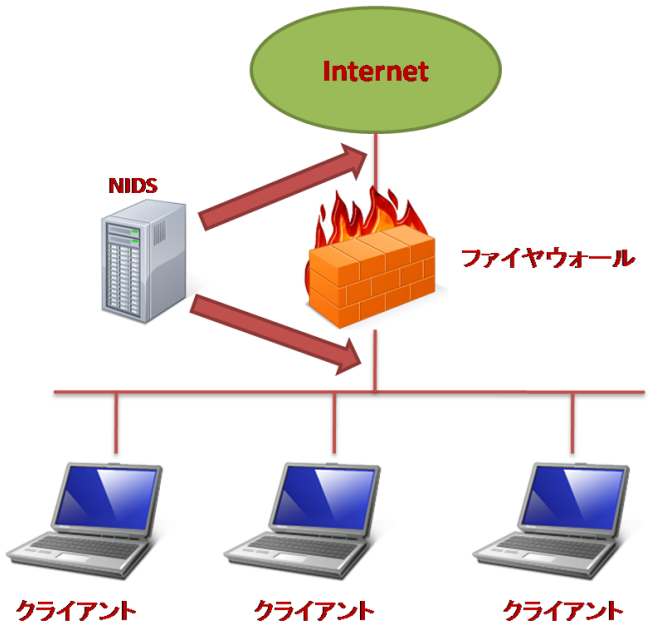


図 2.5: NIDS の構成図

- アプリケーションに対する攻撃を検知できない
  SQL インジェクション [10] をはじめ，組織がサイト用に独自の仕様で開発したアプリケーションプログラムの脆弱性を突いた攻撃などはほとんど検知できない．
- 誤検知
  第 2.1.1 項で前述した IDS の誤検知によって，NIDS における誤検出の割合を測る指標として，フォールスポジティブとフォールスネガティブがある．フォールスネガティブとは異常な行為が発生しているにもかかわらず，それを不正として検出できないことを指し，フォールスポジティブとは正常な行為を不正として検出してしまう誤検出のことを指す．

[11] によって，イベントを検知した後で，表 2.1 に示すようなアクションを実行することが可能がある．

また，NIDS と HIDS を主に検知機能で比較し，表 2.2 にまとめておく．

本研究では，NIDS のログをチューニングすることを目指す研究のため，本論文でここから，書いてある「IDS」は NIDS を暗示される．

分類	応答内容	NIDS	HIDS
通知・記録機能	管理用コンソールへのアラート通知	○	○
	指定されたアドレスに対してメールで通知	○	○
	指定されたホストに SNMP トラップを送信	○	○
	ログ出力（ローカル環境）	○	○
	Syslog サーバへのメッセージ送信	○	○
プログラム実行・停止機能	指定されたプログラムの実行又は停止	○	○
ファイルやレジストリの復元	変更が確認されたファイルやレジストリを保存しているオリジナルの状態に復元	×	○
セッション切断，接続制限機能	TCP コネクションの切断（RST パケットの送信）	○	×
	UDP, ICMP の遮断（ICMP port unreachable の送信）	○	×
	ファイアウォールの ACL を動的に変更して防御	○	×
	アカウントのロックアウト，ログインの拒否，上位権限への昇格制限	×	○
	特定のファイルへのアクセス制限	×	○
	受信した特定のパケットの破棄	×	○
その他その他	検知した通信に対する応答	○	×

表 2.1: IDS がイベント検知後に実行するアクション

表 2.2: NIDS & HIDS を検知機能で比較

項目	NIDS	HIDS
導入方法	監視の対象となるネットワークセグメントに接続（専用のハードウェアが必要）	監視の対象となるホストにインストール（専用のハードウェアは不要）
監視方法	流れているパケットを監視（暗号化されたパケットは監視不可）	常駐しているホスト上で行われている操作やポートの状態などを監視
ポートスキャン	○	○
BOF 攻撃	○	○
DoS 攻撃	○	△
サーバプログラムへの各種コマンドの発行	○	×
ログインの成功・失敗	△	○
SQL インジェクション	△	△
重要なリソースへのアクセス	×	○
不正なプログラムのインストール	×	○
Web コンテンツの改ざん	×	○
メールによるファイルの流出	×	○
攻撃の排除方法	×	×

△ は一部検知可能を意味する

2.1.3 分析方法による分類

侵入検知システムは大きく不正検出（Misuse Detection）という手法を用いたものと，異常検出（Anomaly Detection）という手法を用いたものの 2 つに分類することができる．

1. 異常検出（Anomaly Detection）
   システムやユーザーの正常時における振る舞いを記録したプロファイルデータを用意しておき，このデータと取り込んだパケットを比較し，大きく異なったものを異常として検出する手法が異常検出である．不正なことを直接検出するのではなく，通常と異なる行為を検出するので，まずユーザのシステム利用のログイン時刻や使用時間など，ネットワークトラフィックの状況から，正常な振る舞いを定義しなければならない．これ以後，システムやユーザの振る舞いを監視し，正常状態から逸脱するときに警告を発する．例えば，“ 1 秒当たりに 5MB の通信量が存在する状態 ” が平常時であると定義されており，または“ それを超過した場合は異常と見なす ” と定義されていた場合，もし 1 秒間に通信量が 10MB であれば IDS は異常であることと判断し，警告を発する．

   異常検出という手法によって，検知可能な事象には次のようなものがある．

   - プロトコルの仕様に反したデータの流れ
   - プロトコルの仕様に従っていないヘッダ情報を持つパケット
   - 大量に発行されたコマンド　など

   図 2.6 にみて，異常なものを検出手法が単純に分かることになる．

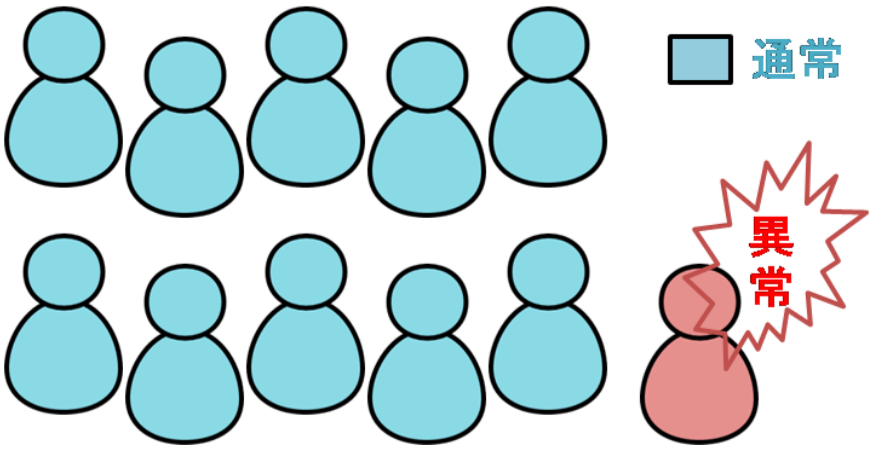


図 2.6: 異常検出のイメージ

この方法では，異常な行為を見つけるものにしたがって，未知の攻撃（zero-day attack[12]）や正常なパケットを大量に送りつけるタイプの DoS 攻撃 [2] などに対応することが

可能である．しかし，システムの変動やユーザの操作などに大きく影響を受けるという欠点があり，正常なのに異常として警告をする誤検知（フォールスポジティブ）の発生確率が高いといわれている．一般に異常検出の手法を用いた侵入検知システムは，システムやユーザの振る舞いの変化を検出するために，侵入以外の原因による状態の変化も異常として検出してしまう．このために，侵入がないときに侵入があるのように警告を発してしまう．フォールスポジティブをはじめ，異常検知システムを実用するために，以下の問題点を解決する必要がある．

- 正常プロファイルの作成の問題
  異常検出において，正常な状態と異常な状態を区別するという技術があり，なにを「正常」と定義するのかという問題がある．正常プロファイルを作成している時に不正のことが行われていないのを保証するのは難しい．そのうえで，大学をはじめ，変動が激しい使い方が想定されるシステム管理者の環境において不正の区別が困難であるという問題も存在する．
- しきい値など各種パラメータを設定する手法の問題
  異常検出手法を用いた侵入検知システムを実用化にするために，さまざまなパラメータによって検出精度をチューニングできる．これらのパラメータは，誤警報数の推移を見極めつつ経験的に設定し，本章で紹介したようなアルゴリズムによる自動設定によって決定する．ただし，このようなパラメータに関する一般的な手法を確立するという困難がある．
- 学習データが操作される可能性
  順次にユーザプロファイルを学習するアノマリ型侵入検知システムは，意図的にプロファイルを操作される危険性があることに注意しなければならない．たとえば，通常午前中にログインするユーザのアカウントを不正に使用する侵入者が深夜にログインしたら，異常として警告を発するが，毎日ログイン時刻を変化させていった場合，学習するようなシステムは不正なログイン時間を検出するのが難しい．

2. 不正検出（Misuse Detection）
   不正検出とは，あらかじめデータベースに定義しておいた攻撃・侵入行為の特徴と照合して攻撃・侵入を検出する手法である．あらかじめ登録する特徴情報のことをシグネチャと呼ぶこともあるため，シグネチャベース検出と呼ぶこともある．シグネチャは，数値，文字列，ルールなどの形態をとる．基地の攻撃パターンのシグネチャを前もってデータベースに蓄積しておき，ネットワークからの入力情報と合致するシグネチャがデータベースの中で見つかる場合には，”攻撃あり”と判断できる．エキスパートシステムのルールをシグネチャとして利用する例として，図 2.7 に一般的バッファオーバーフロー攻撃検知用のルールを示す．このルールは異常に長いパラメータを持つ EXEC コマンドの実行を監視しており，それを検知した時点で ALERT メッセージを発する．

   不正検出手法を用いた侵入検知システムの多くが，パターンマッチングの手法を用

```
1 rule[BSM_LONG_SUID_EXEC(*):
2 [+e:bsm_event]
3 [? |e.header_event_type == ‘AUE_EXEC ||
4 e.header_event_type == ‘AUE_EXECVE]
5 [? |e.subject_euid != e.subject_ruid]
6 [? |contains (e.exec_args, ”ˆ\\”) == 1]
7 [? |e.header_size>‘NORMAL_LENGTH]
8 ==>
9 [! |printf(”ALERT: Buffer overrun attack \
10 on command %s\n”, e.header_command)]
11 ]
```

図 2.7: 一般的バッファオーバーフロー攻撃検知用のルール

いている．パターマッチングは，入力データにルールやシグネチャによって記述されたパターンと同一なものが含まれているか検索する処理を指す．パターンマッチングによる攻撃検知のイメージを図 2.8 に示す．

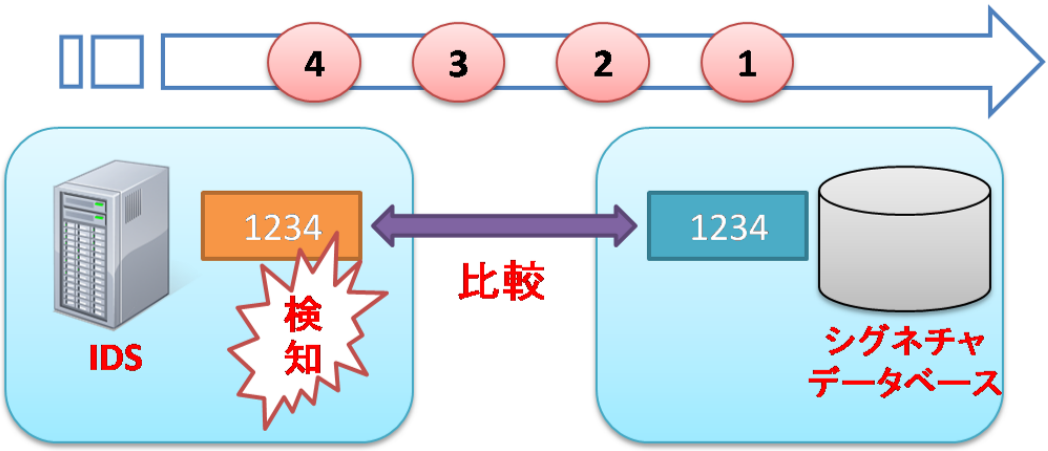


図 2.8: パターンマッチングによる攻撃検知のイメージ

パターンマッチングによって検知可能な事象には，次のようなものがある．

- ポートスキャン
- 脆弱性検査ツールによるスキャン
- OS やサーバソフトウェアの既知の脆弱性を突いた攻撃（BOF 攻撃 [13] など）
- ネットワークを通じて行われるパスワードクラッキング　など

不正検出は，シグネチャによる攻撃を判断するため，シグネチャに登録されていない攻撃パターンに対して，異常なのに正常として判断し，警報を発しない誤検知（フォールスネガティブ）を引き起こす．未知の侵入や攻撃を検知することができないため，常に新しい侵入方法に対応したシグネチャをデータベースに追加する必要がある．したがって，既知の侵入パターンをすべて検出するためには，膨大なシグネチャを必要とする．ただし，シグネチャの増加に比例して IDS の処理に負担がかかることである．また，不正検出は，トラヒックの急激な増加，偽装通信などの検知も困難である．

不正検出の特徴は次のとおり．

- リアルタイムで検出が可能
  ネットワークの入力情報をリアルタイムでパターンと比較するため，侵入を検出するまでのタイムラグが発生しない．
- 侵入検知システムによってルールの記述方法が異なる
  監視対象の環境に応じたユーザー定義のルールを記述する場合には，侵入検知システムによって記述方法が異なることである．そのため，侵入検知システムに応じたシグネチャの記述方法をユーザー習得する必要がある．また，一部の侵入検知システムにはルールの記述方法が非常に難解なものがある．
- サイト独自のアプリケーションの脆弱性を突いた攻撃の検出が不可能
  一般的に使用されている侵入検知システムは非常に多くのシグネチャをもっているが，それらは手法や仕様がある程度公になっている市販製品や，広く使われているシェアウェア，フリーウェアなどに対する攻撃パターンである．したがって，組織がサイト用に独自の仕様で開発したアプリケーションプログラムの脆弱性を突いた攻撃などをパターンマッチングによって検知することはできない．
- 異常検出より誤検知率が低い
  侵入検知システムが捕捉しようとする特定の行為を EOI と呼ぶことがある．不正検出は，検出する EOI を利用者が規定し，その規定にしたがってパターンを記述する．そのため，パターンが適切に記述されていれば，利用者の意図しない EOI を検出することはない．そのように，異常検出に比べてフォールスポジティブが低く抑えることができる．

2.2 IDS 運用における課題

- リスク評価できない警告イベントの調査負担
  IDS が検知する警告イベントの多くはそのリスクを評価することが難しい点である．リスクが高いと考えられる警告イベントの発見が，実際に被害をおよぼすインシデントの発見へと撃がる．ただし，既存の IDS が出力する警告イベントには，リスク評価のための十分な情報が付与されていない．りすくを判断できないイベントについては，実地調査などによってリスク評価する方法が挙げられるが，管理者の負担が大きくなってしまう．

- 膨大な警告イベントの状況把握の負担
  NIDS が検知する警告イベントの数は膨大であり，全ての警告イベントをネットワーク管理者が 1 つずつ解釈し全体像を把握するのは不可能である．全体像が把握できなければ，イベントの発生パターンや発生頻度の変化を認識できない．パターンや頻度の変化は，通常リスク評価ができないイベントや誤検知の多いイベントであっても，注意すべきインシデントであることを示す場合が多く，認識できなければインシデントを見落としてしまう可能性がある．

2.3 実環境における IDS の警告イベントの検知数

侵入検知を運用する場合，誤検知による大量ログの出力が大きな課題となっている．たとえば，500 台のホストが接続された IPv4 ネットワークを Snort[14] で監視した場合，1 日平均のログ出力数 68401 件で，検知数が多い日は 10 万件を超えている事が報告されている [15].

慶応大学湘南藤沢キャンパスでの研究室ネットワーク環境を監視している Snort を用いて，2011 年 1 月 1 日から 1 月 7 日までの警告イベントの検知数について調査したところ，結果は表 2.3 に示す.

検知数は最も多い日で 490504 件，最も少ない日でも 96496 件と 1 日平均 203350 件で，膨大な数となっている．これら全てを管理者に警告したとすると，管理者がすべてを確認することは不可能で，また大きな負担となってしまうことは明らかである.

検知数は 1 日平均 203350 であるが，警報したシグネチャ数は 1 日平均 183 であることによって，複数個検知されたシグネチャが多数ある．図 2.9 と図 2.10 に見ると，それを確認できた．2011 年 01 月 01 日にも 01 月 02 日にも，検知されたシグネチャ数と検知数が違ったことがあるが，4 つのシグネチャの検知数は一日中全ての検知数約 80 %である.

日付	全ての検知数	シグネチャ数
2011 年 1 月 1 日	490504	166
2011 年 1 月 2 日	289206	162
2011 年 1 月 3 日	107627	160
2011 年 1 月 4 日	96496	185
2011 年 1 月 5 日	125876	189
2011 年 1 月 6 日	106712	198
2011 年 1 月 7 日	207025	218

表 2.3: 研究室ネットワーク環境での検知数とシグネチャ数

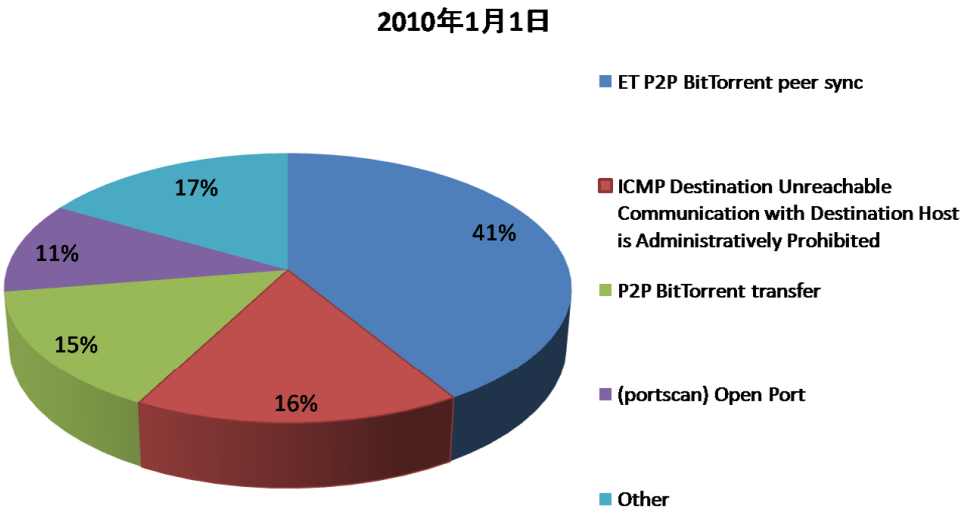


図 2.9: 2010/01/01 の警告イベント

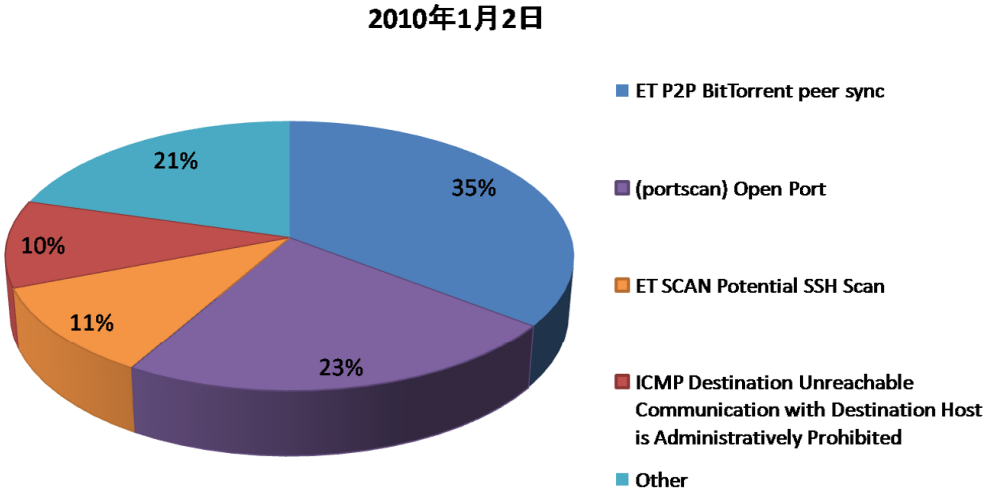


図 2.10: 2010/01/02 の警告イベント

NIDS が検知するセキュリティイベント数はネットワークのアドレス範囲や構成によって変化するが，基本的には接続しているホスト数に比例して増加する傾向がある．より多くのホストが接続している大規模なネットワークでは，さらに膨大な数量の警告イベントが検知されると予想される．

2.4 本論文の着眼点

第 2.3 節でのべた，現在の NIDS を用いて，毎日膨大な検知数が出力されるため，すべての検知イベントを判断することは不可能である．本論文は，管理者の負担が少なくなるように NIDS のログを調整することを目的として，すべての検知イベントを警告せずに，異常な検知だと判断したもののみ管理者へ警告する手法を提案する．フォールスポジティブと正検知に分けず，正常な警告イベントと異常な警告イベントに分ける手法である．警告イベントの過去の傾向に基づいて，異常検知数を出力したものを判断することが可能である．異常な警告を検出するために，フォールスポジティブだけではなく，意味がない攻撃（被害なしの場合）を除去できる．それで，直接に攻撃を検出しないが，攻撃を検出するための時間が短くなると考えている．

2.5 まとめ

本章では，侵入検知システムの基本的なことについて挙げた．検知対象空間による分類し，ネットワーク型侵入検知システムとホスト型侵入検知システムがあり，分析方法による分類すれば，アノマリ型侵入検知システムとシグネチャ型侵入検知システムがある．侵入検知システムのログが多すぎるため，管理者の負担が大きい．だから，過去のログを学習し，できるだけ異常なログを明示的に示す必要がある．

第3章　関連研究

本章では，本研究との関連研究とするIDSのログ分析に関する既存手法を述べる．

3.1　警告の相互関係を築く手法

Frederic CuppensとAlexandre Miegeの論文Alert Correlation in a Cooperative Intrusion Detection Framework[16]で，警告の相互関係を築く手法について述べている．現在のネットワーク型侵入検知システムはネットワークに流れるパケットを監視し，不正なアクセスを検知するたびに管理者へ警告する．しかし，全部の警告は初等の警告だと思い込むことがあったから，警告を組み立てて，グローバル的な警告を生成することを目標としてCRIMと呼ばれるIDSのモジュールを設定している．CRIMの構成を図3.1に示す．

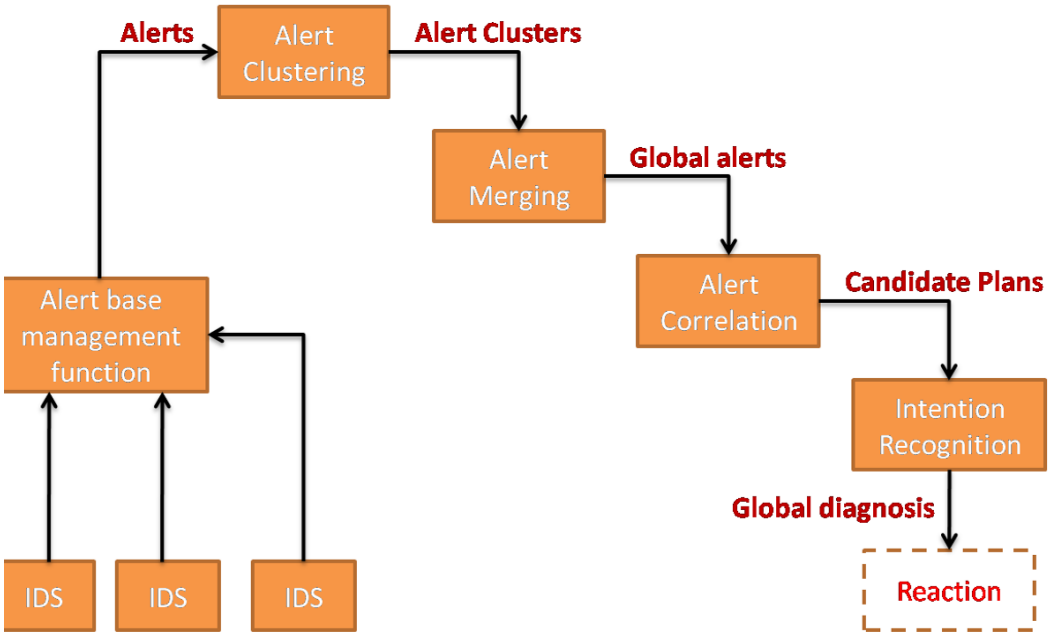


図 3.1: CRIM構成

まず，多少のIDSの警告を受け取り，データベースに蓄積する．それから，クラスタリング機能はデータベースをアクセスし，警告をかたまりにする．マージ機能は1つの警告クラスターごとに1つの新たらしい警告を作成する．その新しい警告は警告クラスタの

全部の情報を持っている．最後，マージ機能に出力された警告の相互関係を築くことにより，基本の警告ではなく，総合的な警告を設定している．

そのうえ，警告の相互関係を築くことにしたがって，IDS の警告数を減らすことができる．

3.2 適応的な警告分類手法

フォールスポジティブを減らすことを目的として，Tadeusz Pietraszek の論文 [17] で ALAC（Adaptive Learner for Alert Classification）が言及されている．ALAC は警告をフォルスポジティブと正検知に分類し，分析結果を管理者へ通報する．管理者からフィードバックにしたがって，ルールを更新し，新しい警告を分類することができる．ALAC の構成を図 3.2 に表示する．

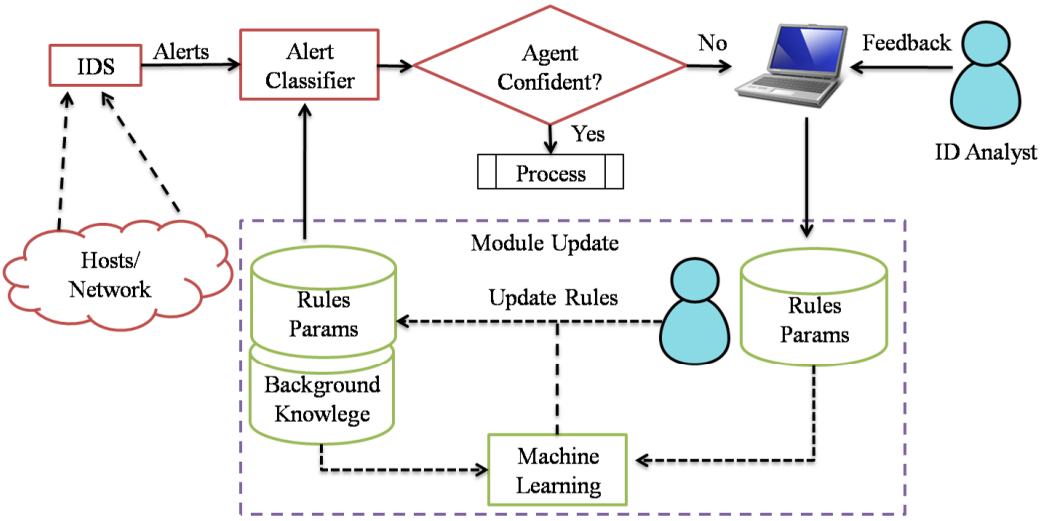


図 3.2: ALAC 構成

論文での提案手法は正検知を判断する手法とフォールスポジティブを判断する手法がある．フォールスポジティブを確認するために，データマイニング技術 [18] を用いて root cause analysis[19] あるいは統計的なプロファイリング [20] などを利用する．

ALAC の分類ルールを調整するのは管理者からフィードバックが必要であるが，以下の理由のため，管理者は自分で分類ルールを書かないほうがいい

- 管理者の知識の限界
  フォールスポジティブとする警告を個々に確認できる管理者にとっても，フォールスポジティブの特徴づけるルールを設定することは簡単ではない．

- ダイナミックなネットワーク環境
  実環境に警告の特徴が変動的なものとともに警告分類ルールを常に調整する必要がある.

フォールスポジティブとする警告を自動的に消去できる ALAC の評価によって，ALAC を用いてフォールスポジティブを 30%減らす.

3.3 誤検知低減の研究動向

不正侵入検知の誤検知低減技術について，既存の研究動向を紹介する論文 [21] である. この論文によって，侵入検知システムの研究は，最近盛んに取り込み始められるようになったが，現状は個別技術に対する研究が主体で，侵入検知全体の総合的な研究はまだ少ない．現状の侵入検知システムの課題としては，

- フォールスポジティブの低減
- フォールスネガティブの低減
- IDS 熟練運用者の確保
- パフォーマンスの確保
- 導入・運用コストの低減
- 暗号通信への対応

などが挙げられた．特に，フォールスポジティブとフォールスネガティブ，すなわち誤検知については，実運用上既に大きな問題となっている．誤検知の主な原因は，シグネチャ設定の難しさ，プロファイルの無効化，IDS 設定チューニングの難しさ，シグネチャ更新の煩雑さ，ログに対する不適切な統計処理，IDS のパフォーマンス不足であることを述べ，これら課題に対する研究成果の一部をこの論文には紹介する．また，誤検知低減以外の IDS の今後の研究課題として，インシデントのカウント基準，IDS の評価方法，IDS 熟練運用者の確保について述べ，侵入検知研究の総合的な推進の必要性について述べた.

3.4 まとめ

本章では，複数の IDS ログ分析手法について述べた．IDS の誤検知を低減するために、警告イベントの中にフォールスポジティブと正検知を分類手法もあるし、警告イベントをクラスタリングし、相関関係を調査し複数の警告イベントから 1 つの警告を作成する方法もあった。

第4章　提案ログ分析手法

本章では，管理者の負担を減らすために警告イベントの過去の傾向に基づく IDS のログ分析手法を提案する．提案するログ分析手法は，一定の期間の警告をクラスタリングした後で，警告イベント毎に過去の傾向を分析し，それに基づいて警告イベント毎にアノマリスコアの算出にしたがって，ラベル付けをする．その上で，管理者は警告の有無のしきい値を自分で選定することによって，警告の一部分だけを見せることができる．以下，クラスタリング方法，警告イベントの過去の傾向の分析方法，アノマリスコアとラベル計算方法，しきい値の設定について順に述べる．

4.1　警告イベントをクラスタリングする手法

警告イベントの数は膨大であり，大量の検知ログが出力されるため管理者が常に監視を行うことは困難である．また，警告イベントが検知されるたびに管理者へ警告していては，管理者は大量の警告をすべて確認しなければならず大きな負担となる．そこで，一定の期間 t の警告イベントをクラスタリングすることで，警告数をある程度削減することを考える．第 2.3 節に述べたことによって，短時間で複数個検知されたシグネチャがあり，そこで，期間 t で検知された警告イベントについて，シグネチャ名にクラスタリングすることにする．これにより，期間 t で検知された警告イベントを集約することができるので，警告イベントの数をある程度抑制することができ，結果として管理者の負担を減らすことができると考えられる．

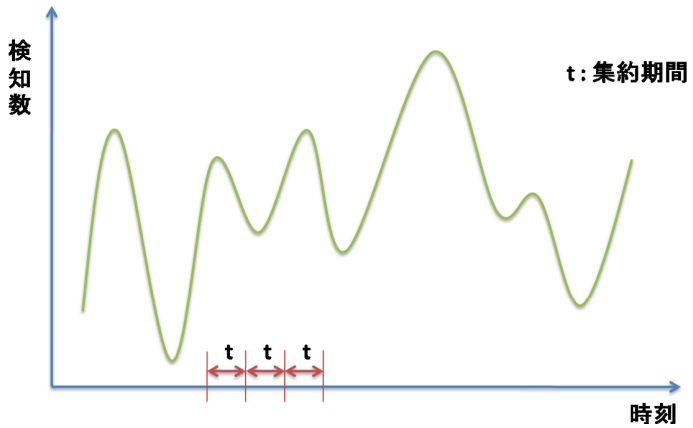


図 4.1: 集約期間 t

図 4.1 によって，例えばデータ取得期間を 2011 年 1 月 1 日 00 時 00 分 00 秒から 2011 年 1 月 1 日 23 時 59 分 59 秒までとし，もし期間 t を 1 日として，警告を表 4.1 のように示し，あるいは，期間 t を 6 時間として，警告を表 4.2 のように示す.

Date	Time	Signature	Occurence times
2011/01/01	00:00:00 - 23:59:59	Signature A	a
		Signature B	b
		Signature C	c

表 4.1: クラスタリングされた警告（集約期間 t = 1 日）

Date	Time	Signature	Occurence times
2011/01/01	00:00:00 - 05:59:59	Signature A	a_1
		Signature B	b_1
		Signature C	c_1
	06:00:00 - 11:59:59	Signature C	c_2
		Signature D	d_1
		Signature E	e_1
	12:00:00 - 17:59:59	Signature A	a_2
		Signature B	b_2
		Signature F	f
	18:00:00 - 23:59:59	Signature B	b_3
		Signature D	d_2
		Signature E	e_2

表 4.2: クラスタリングされた警告（集約期間 t = 6 時間）

4.2 過去の傾向を分析する手法

本節では，シグネチャごとに警告イベントの過去の傾向を分析するために，3 シグマ規則を用いてデータの分布を判断する方法について述べる.

4.2.1 過去期間 T_p

対象とするシグネチャの過去の傾向を把握するために必要な過去期間 T_p を定める．過去期間 T_p における集約期間 t の警告イベントをシグネチャ名にクラスタリングし，各シグネチャ毎の検知数のばらつきを分析する．たとえば集約する期間 t を 1 日，過去期間 T_p を 30 日とし，2011 年 01 月 01 日のログ（00 時 00 分 00 秒〜23 時 59 分 59 秒）を分析する場合，シグネチャごとに 2010 年 12 月 2 日 00 時 00 分 00 秒から 2010 年 12 月 31 日 23 時 59 分 59 秒までの毎日（00 時 00 分 00 秒〜23 時 59 分 59 秒）検知数を取ることにしたがって，傾向を判断する．(図 4.2)

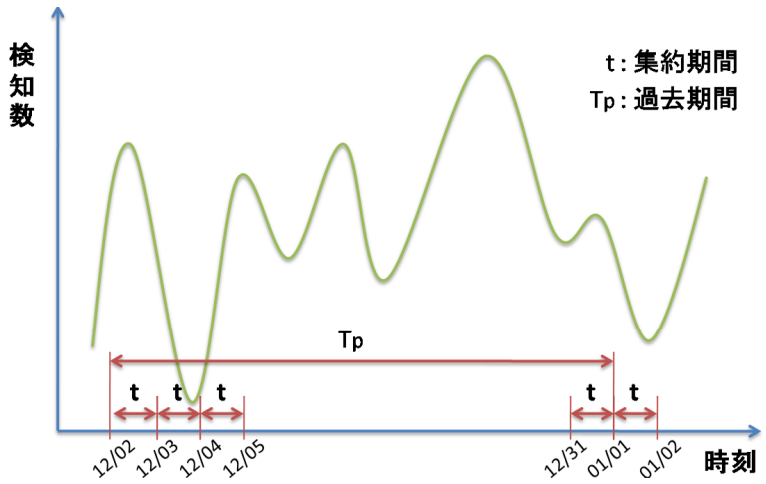


図 4.2: 過去期間 T_p

検知数のばらつきを見るために，集約する期間ごとに検知数の平均と標準偏差を求める．それから，3 シグマ規則を適用し，過去期間 T_p における各シグネチャの検知数の分布を判断することができる．

4.2.2 3 シグマ規則

1. 3 シグマ規則とは，データの正規分布について統計的な規則である．68-95-99.7 規則と呼ばれていることもある．それを図 4.3 に示す．

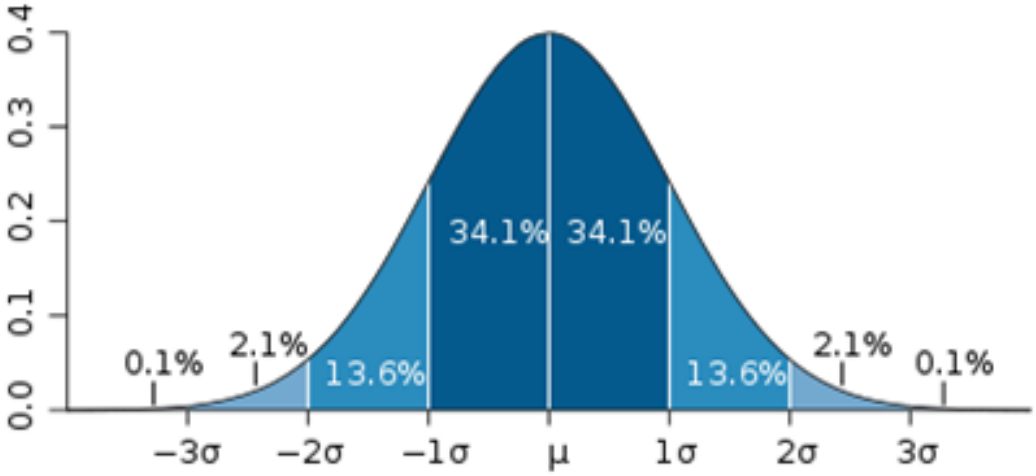


図 4.3: 3 シグマ規則

図 4.3 をみて，以下のことが分かるになる．

- 価値の約 68%は平均の 1 つの標準偏差の内にある．
- 価値の約 95%は平均の 2 つの標準偏差の内にある．
- 価値ほとんどすべては (実際に，99.7%) 平均の 3 つの標準偏差の内にある．

つまり，確率変数 X が $N(\mu, \sigma^2)$ に従う時（μ：平均，σ：標準偏差），平均 μ からのずれが ±1 σ 以下の範囲に X が含まれる確率は 68.26%，±2 σ 以下だと 95.44%，さらに ±3 σ だと 99.74% となる．

2. 3 シグマ規則の適用を実際に考証する
3 シグマ規則の確実性を見るために，研究室のネットワークを実験環境として Snort を用いて，2010 年 12 月 02 日から 2010 年 12 月 31 日まで（30 日間）の警告イベントの検知数について調査し，平均検知数が大きく異なる 2 つのシグネチャ（portscan と SNMP trap UDP）を取り上げる．これら 2 つのシグネチャの，2010 年 12 月 02 日から 08 日まで（1 週間）の各日ごとの検知数を表 4.3 に示す．

また，2010 年 12 月 02 日から 12 月 31 日（30 日間）においてその 2 つのシグネチャの検知数を図 4.4 と図 4.5 に示す．

	シグネチャ	
日付	（portscan）Open Port	SNMP trap UDP
2010/12/02	55953	510
2010/12/03	65429	452
2010/12/04	18089	307
2010/12/05	6708	158
2010/12/06	87341	483
2010/12/07	69131	447
2010/12/08	19298	579

表 4.3: 2 つのシグネチャの各日毎の検知数

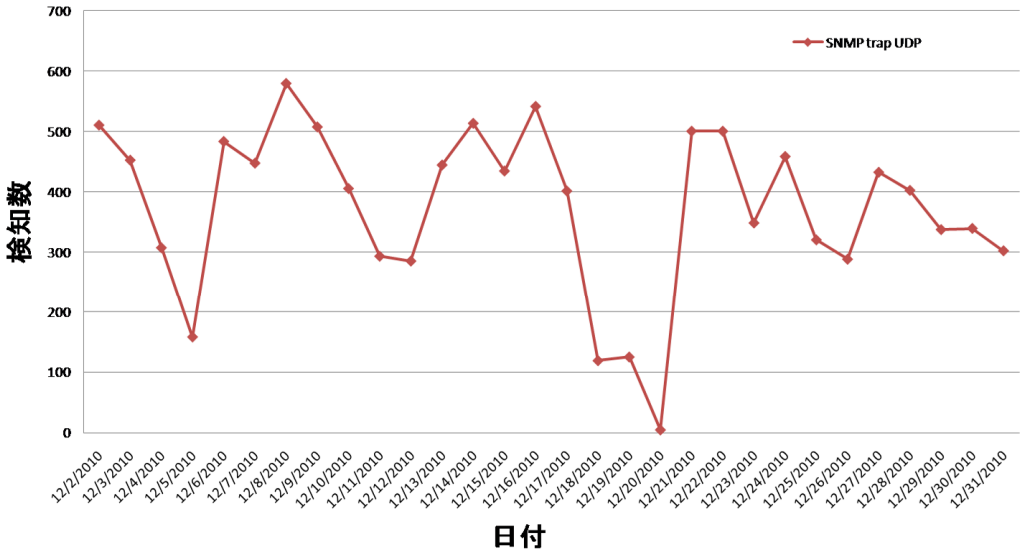


図 4.4: （portscan）Open Port の検知数

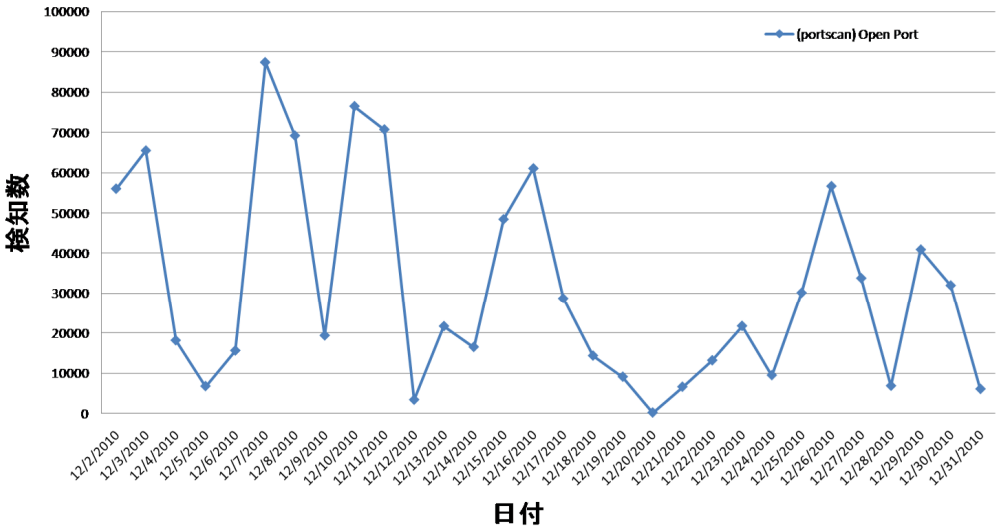


図 4.5: SNMP trap UDP の検知数

図 4.4 と図 4.5 を見て，2 つのシグネチャの平均検知数が全く違うことが分かる．ただし，2 つのシグネチャの過去の傾向を判断するために平均 μ と標準偏差 σ を計算し，分布を調査する時に，図 4.6 と図 4.7 をみて，2 つの結果はだいたい同じであることが分かる．検知数はほとんどは（μ − σ，μ + σ）の範囲にあり，(μ − 2 × σ，μ + 2 × σ）以外の検知数は少ない．

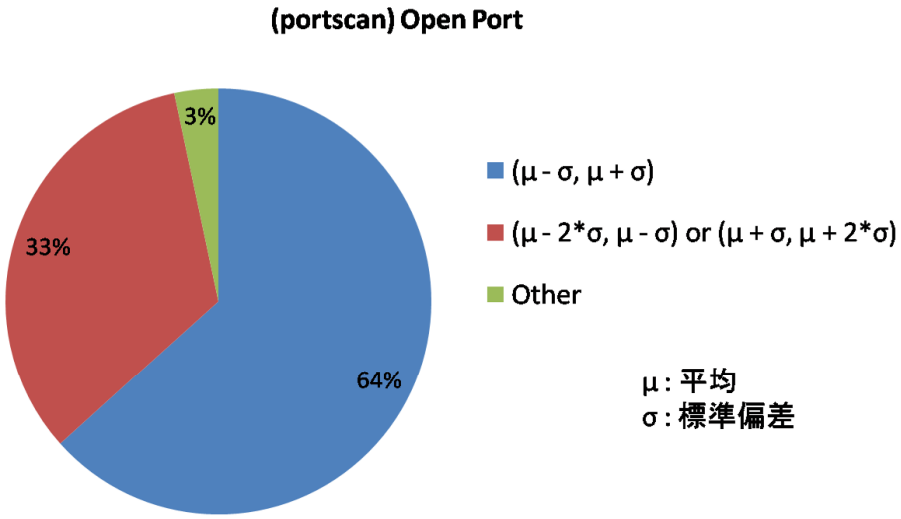


図 4.6: （portscan）Open Port の検知数の分布

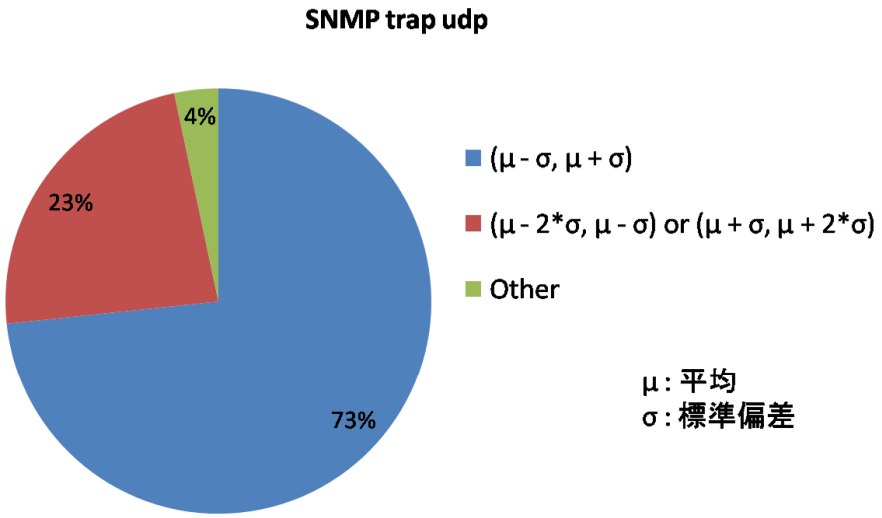


図 4.7: SNMP trap UDP の検知数の分布

調査結果によって，3 シグマ規則を実際のデータに対して適用できると考える．3 シグマ規則によってシグネチャごとに過去の検知数の分布をすることにしたがって，警告イベントの過去の傾向を分析する可能がある．

4.3 アノマリスコアとアノマリラベルの設定

過去期間における集約期間の警告イベントをクラスタリングし，3 シグマ規則を適用し警告イベントの過去の傾向を分析するため，警告イベントを監視し異常な検知数かどうかを判断するためにアノマリスコアを設定する．そして，アノマリスコアにしたがって，警告ごとにアノマリラベルを付ける．

4.3.1 アノマリスコアの設定

集約期間を t における検知数を x，過去期間を T_p，平均検知数を μ，標準偏差を σ とするとき，アノマリスコア s_a は以下の式で定義される（図 4.8）

$$s_a = \frac{|x - \mu|}{\sigma}$$

図 4.8: アノマリスコアの式

平均 μ と標準偏差 σ は以下の式で定義される（図 4.9 と図 4.10）

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

図 4.9: 平均検知数の式

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$$

図 4.10: 標準偏差の式

3 シグマ規則とアノマリスコアの式を用いて，過去の傾向に基づいて異常な検知数を検出したか，正常な検知数を検出したか判断する可能がある．アノマリスコアと警告イベントの過去の分布の関係を図 4.11 に示す．

アノマリスコアを s_a とするときに

- s_a <1
  図 4.11 の**範囲 1** にある場合，すなわち過去期間における約 **68%**の検知数の範囲にある．

- 1 <s_a <2
  図 4.11 の**範囲 2_a と範囲 2_b** にある場合，すなわち過去期間における約 **27%**の検知数の範囲にある．

- 2 <s_a
  図 4.11 の**範囲 3_a と範囲 3_b** にある場合，すなわち過去期間における約 **5%**の検知数の範囲にある．

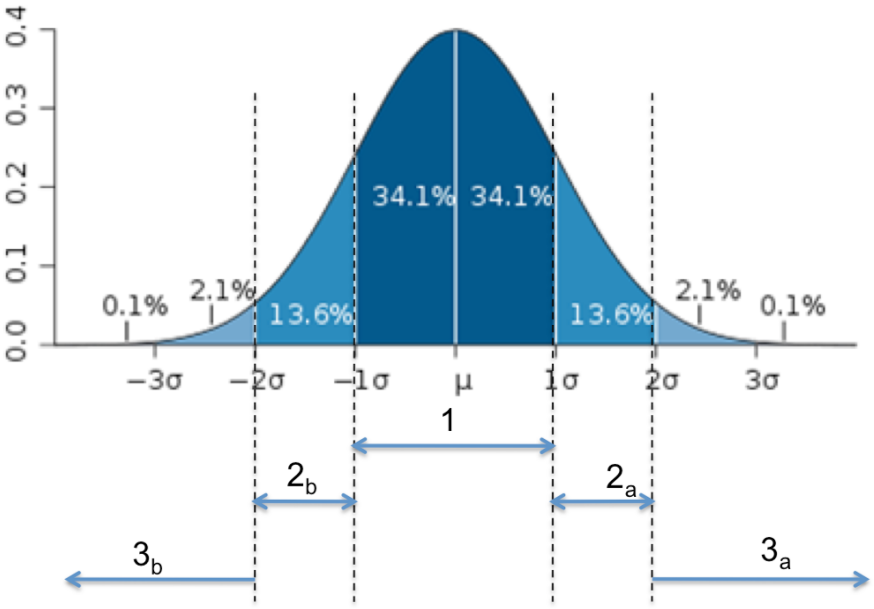


図 4.11: アノマリスコアと 3 シグマ規則

警告イベント各によって，平均と標準偏差が大きく異なることである，ただし，アノマリスコアは警告によらず警告の異常的を判断できる係数である．

4.3.2 アノマリラベルの設定

アノマリスコアによって，アノマリラベルを表 4.4 に定義される．

アノマリスコア	アノマリラベル
$s_a < 1$	1
$1 \leqq s_a < 2$	2
$2 \leqq s_a$	3

表 4.4: アノマリラベルの定義

アノマリラベルはアノマリスコアに正比例の関係にある．レベル 3 のシグネチャは異常なので，ラベル 3 からラベル 1 までの順序をきめることにより，管理者はどんな警告イベントに注意したほうがいいが分かると考える．

4.3.3 しきい値の設定

全部の警告イベントを確認しなく，一部分の正常な警告を無視し異常なものだけ見るために，しきい値を設定しなければならない．具体的に，アノマリスコア s_a がしきい値 s_th を超えないときに，そのシグネチャは異常ではなく集中する必要がないものとして管理者にみせないことである．表 4.5 に示す.

$\mathrm{s}_a < \mathrm{s}_{th}$	管理者にみせないシグネチャ
$\mathrm{s}_a \geqq \mathrm{s}_{th}$	管理者にみせるシグネチャ

表 4.5: しきい値

しきい値はネットワーク環境と侵入検知システムの設定によって，違うことがあるから，管理者は自分でしきい値を決める可能があればいいと考える.

4.4 まとめ

本章では，警告イベントの過去の傾向に基づいて，アノマリスコアとアノマリラベルを警告イベント毎に付けて，警告イベントを監視し，異常な検知数かどうかを判断する手法を提示した．または，しきい値を設定し，全部の警告イベントではなく，一部分の注意するべき警告イベントだけを管理者に見せる方法と提案した．第 5 章ではこれらの手法を実際に検証する.

第5章　実環境における提案手法の適用と評価

本章では，実環境において第4章に述べた提案手法を適用するシステムを設定する．そして，提案手法の性能評価と考察を述べる．

5.1　実験環境

提案手法の実装及び性能評価は，接続ホストが約500台あまりの本学の研究室のネットワーク環境で行っている．
2010年4月から今までネットワークを監視するときに，ネットワーク型侵入検知システムとしてSnortを利用している．
Snortから出力されたログをデータベースとしてMySQL[22]へ保存している．
MySQLからデータを取り出すログ分析部分は警告イベントの過去の傾向を判断し，順番を決めて，結果を管理者に見せる．

5.2　システム設定

システム構成を図5.1と図5.2に示す．

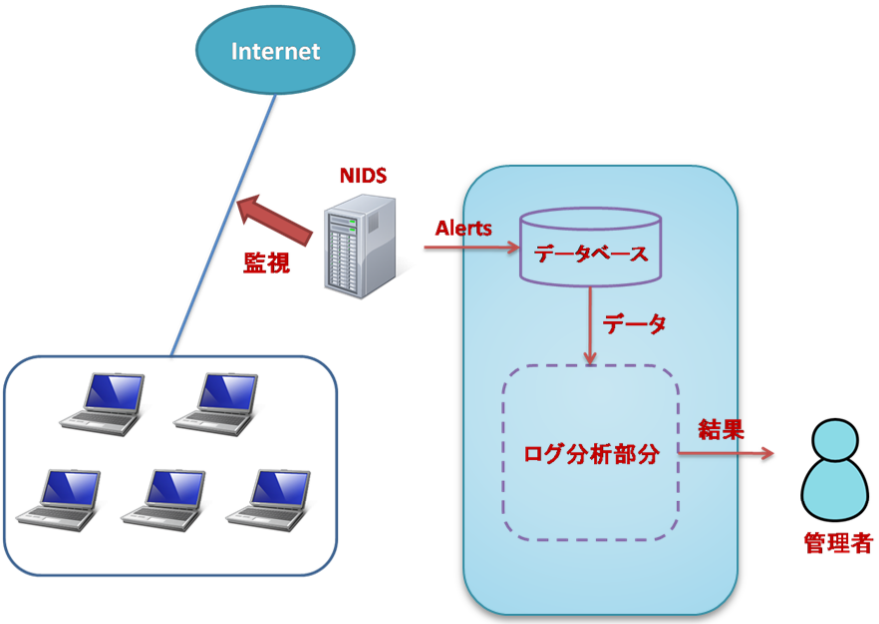


図 5.1: システム構成

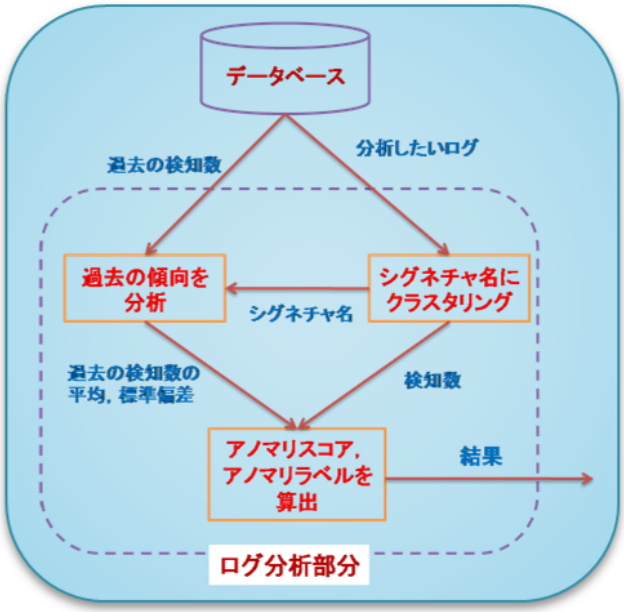


図 5.2: ログ分析部分

不正アクセスを検出する時に，Snort は警告をすることと同時に，MySQL にログを書き込むことである．管理者はログを見たい時刻，過去期間 T_p（学習期間），集約期間 t を入力した後で図 5.3，ログ分析部分は MySQL から過去期間 T_p において集約期間 t の警告イベントをシグネチャ名にクラスタリングし，過去の検知数の分布を調べることにしたがって，警告イベントごとにアノマリスコアとアノマリラベルを付ける．順番を決めた結果は図 5.4 のように示す．

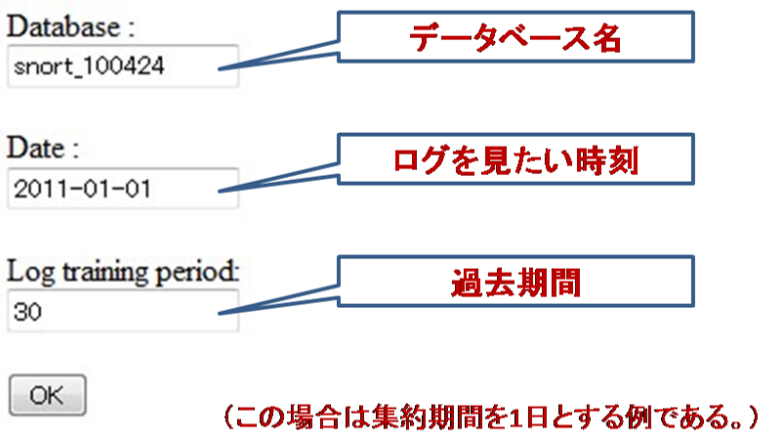
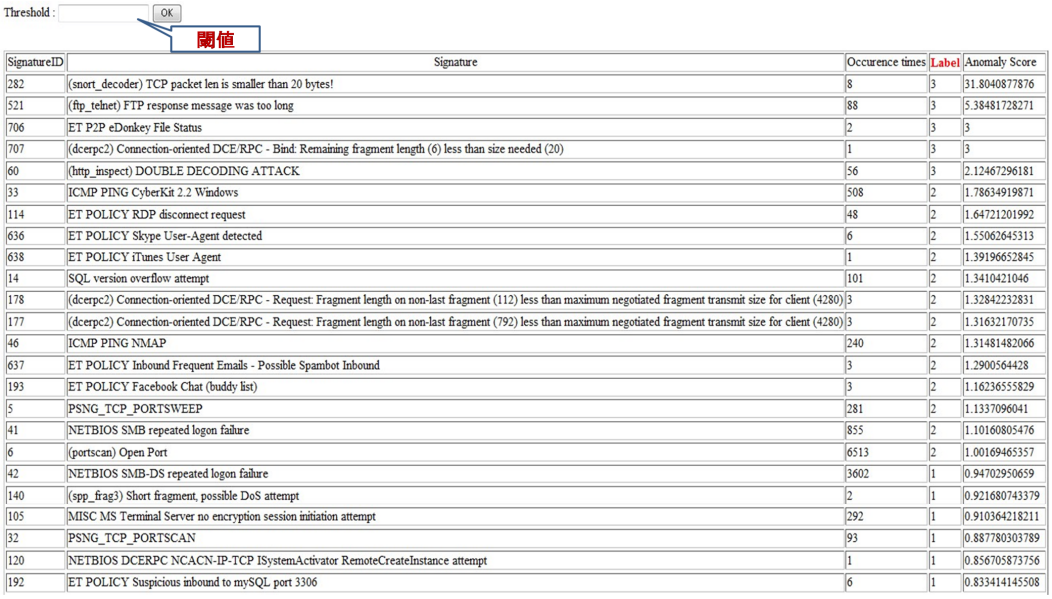


図 5.3: 入力する情報


Threshold : OK 閾値

SignatureID	Signature	Occurence times	Label	Anomaly Score
282	(snort_decoder) TCP packet len is smaller than 20 bytes!	8	3	31.8040877876
521	(ftp_telnet) FTP response message was too long	88	3	5.38481728271
706	ET P2P eDonkey File Status	2	3	3
707	(dcerpc2) Connection-oriented DCE/RPC - Bind: Remaining fragment length (6) less than size needed (20)	1	3	3
60	(http_inspect) DOUBLE DECODING ATTACK	56	3	2.12467296181
33	ICMP PING CyberKit 2.2 Windows	508	2	1.78634919871
114	ET POLICY RDP disconnect request	48	2	1.64721201992
636	ET POLICY Skype User-Agent detected	6	2	1.55062645313
638	ET POLICY iTunes User Agent	1	2	1.39196652845
14	SQL version overflow attempt	101	2	1.3410421046
178	(dcerpc2) Connection-oriented DCE/RPC - Request: Fragment length on non-last fragment (112) less than maximum negotiated fragment transmit size for client (4280)	3	2	1.32842232831
177	(dcerpc2) Connection-oriented DCE/RPC - Request: Fragment length on non-last fragment (792) less than maximum negotiated fragment transmit size for client (4280)	3	2	1.31632170735
46	ICMP PING NMAP	240	2	1.31481482066
637	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound	3	2	1.2900564428
193	ET POLICY Facebook Chat (buddy list)	3	2	1.16236555829
5	PSNG_TCP_PORTSWEEP	281	2	1.1337096041
41	NETBIOS SMB repeated logon failure	855	2	1.10160805476
6	(portscan) Open Port	6513	2	1.00169465357
42	NETBIOS SMB-DS repeated logon failure	3602	1	0.94702950659
140	(spp_frag3) Short fragment, possible DoS attempt	2	1	0.921680743379
105	MISC MS Terminal Server no encryption session initiation attempt	292	1	0.910364218211
32	PSNG_TCP_PORTSCAN	93	1	0.887780303789
120	NETBIOS DCERPC NCACN-IP-TCP ISystemActivator RemoteCreateInstance attempt	1	1	0.856705873756
192	ET POLICY Suspicious inbound to mySQL port 3306	6	1	0.833414145508

図 5.4: 分析した警告

しきい値を入力した後で，しきい値より大きいアノマリスコアがある警告イベントだけを見せる．図 5.5 と図 5.6 のように示す．

Threshold : 1 OK

SignatureID	Signature	Occurence times	Label	Anomaly Score
282	(snort_decoder) TCP packet len is smaller than 20 bytes!	8	3	31.8040877876
521	(ftp_telnet) FTP response message was too long	88	3	5.38481728271
706	ET P2P eDonkey File Status	2	3	3
707	(dcerpc2) Connection-oriented DCE/RPC - Bind: Remaining fragment length (6) less than size needed (20)	1	3	3
60	(http_inspect) DOUBLE DECODING ATTACK	56	3	2.12467296181
33	ICMP PING CyberKit 2.2 Windows	508	2	1.78634919871
114	ET POLICY RDP disconnect request	48	2	1.64721201992
636	ET POLICY Skype User-Agent detected	6	2	1.55062645313
638	ET POLICY iTunes User Agent	1	2	1.39196652845
14	SQL version overflow attempt	101	2	1.3410421046
178	(dcerpc2) Connection-oriented DCE/RPC - Request: Fragment length on non-last fragment (112) less than maximum negotiated fragment transmit size for client (4280)	3	2	1.32842232831
177	(dcerpc2) Connection-oriented DCE/RPC - Request: Fragment length on non-last fragment (792) less than maximum negotiated fragment transmit size for client (4280)	3	2	1.31632170735
46	ICMP PING NMAP	240	2	1.31481482066
637	ET POLICY Inbound Frequent Emails - Possible Spambot Inbound	3	2	1.2900564428
193	ET POLICY Facebook Chat (buddy list)	3	2	1.16236555829
5	PSNG_TCP_PORTSWEEP	281	2	1.1337096041
41	NETBIOS SMB repeated logon failure	855	2	1.10160805476
6	(portscan) Open Port	6513	2	1.00169465357

図 5.5: 警告（しきい値＝ 1）

Threshold : 2 OK

SignatureID	Signature	Occurence times	Label	Anomaly Score
282	(snort_decoder) TCP packet len is smaller than 20 bytes!	8	3	31.8040877876
521	(ftp_telnet) FTP response message was too long	88	3	5.38481728271
706	ET P2P eDonkey File Status	2	3	3
707	(dcerpc2) Connection-oriented DCE/RPC - Bind: Remaining fragment length (6) less than size needed (20)	1	3	3
60	(http_inspect) DOUBLE DECODING ATTACK	56	3	2.12467296181

図 5.6: 警告（しきい値＝ 2）

5.3 評価・考察

5.3.1 過去期間 T_p について

過去期間 T_p における各シグネチャの過去の傾向を分析することのため，期間 T_p に関する考察が必要である．期間 T_p が通常状態であることを前提にしたから，期間 T_p においてある正検知（攻撃）が行えれば，本提案分析手法ではそれを判断できないため，誤った傾向の分析を行ってしまう．ことにしたがって，ある正検知を正常な検知数を示している警告として無視してしまうことになる．その時，フォールスネガティブが増える．しかし，過去期間 T_p に攻撃があっても，被害なしの場合，その攻撃は意味がない攻撃として注意しなくても問題がないから，本研究の提案を適用して，まれな攻撃だけに注意して，攻撃を検出するための時間が短くなる．

問題解決のためには，過去期間 T_p が通常状態であることを判断しなければならず，これはある程度，管理者の能力に依存することになる．

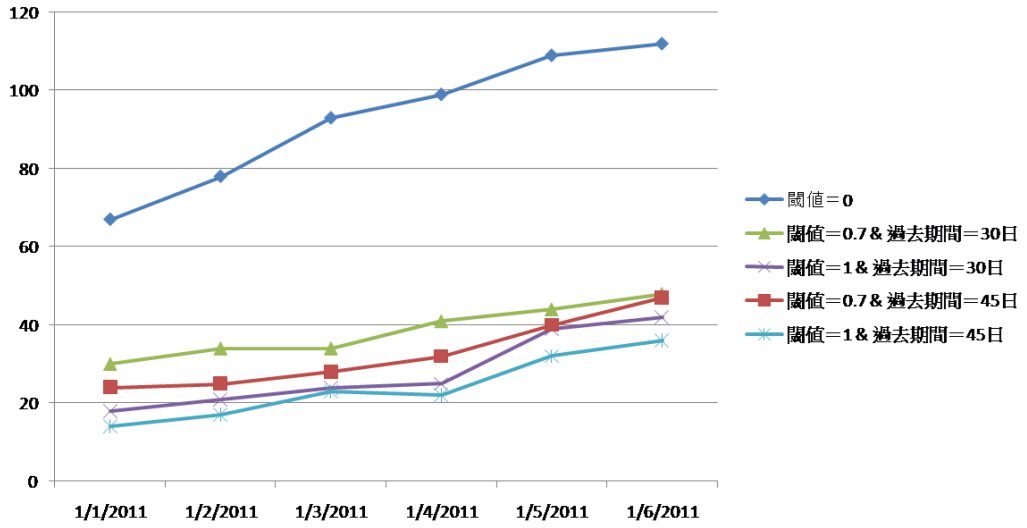


図 5.7: 過去期間 T_p の考察

5.3.2 集約期間 t について

提案分析手法では検知される警告イベントについて集約期間 t の間クラスタリングを行っているため，期間 t を長く設定しすぎると管理者は早急に対応できない．また，短く設定すれば警告イベントのクラスタリングは困難となり，警告数が増えることにより管理者への負担となることが考えられる．どういった指標に着目するかは今後の課題である．

5.3.3 アノマリスコア s_a とアノマリラベルについて

2011 年 01 月 01 日から 01 月 06 日までの警告イベントを分析し，結果は図?? に示す．

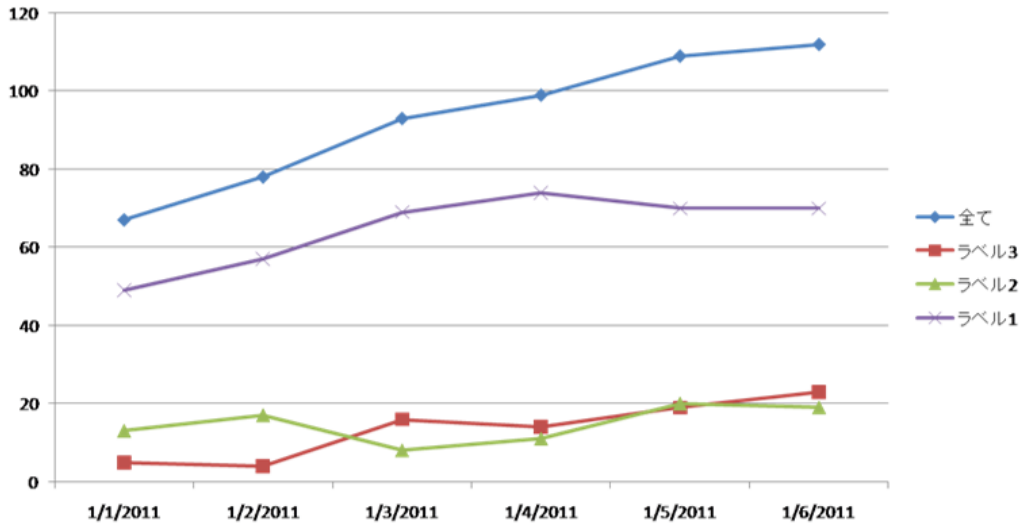


図 5.8: アノマリラベルの実験

休日が終わったときに，研究室に行く人も多くなり，検出した警告も増えることである．図??を見て，毎日検知したシグネチャ数の中に正常な検知数があるラベル 1 のシグネチャが一番多い．しきい値を設定し，ラベル 1 の警告が除去されて，その中にフォールスポジティブの割合を確認しなければならない．もししきい値を 1 としたら，全部のラベル 1 の警告イベントを消して，ラベル 2 とラベル 3 の警告だけ（ラベル 1 より異常性が高い警告）を管理者に見せる時，全て警告の約 40 %以下だけ判断する管理者の負担は減ることである．

5.3.4 しきい値 s_{th} について

しきい値 s_{th} が大きく設定すると，消去される警告イベントが多くなり，フォールスネガティブが増えるかもしれない．また，小さく設定すると，消去される警告が少ないた

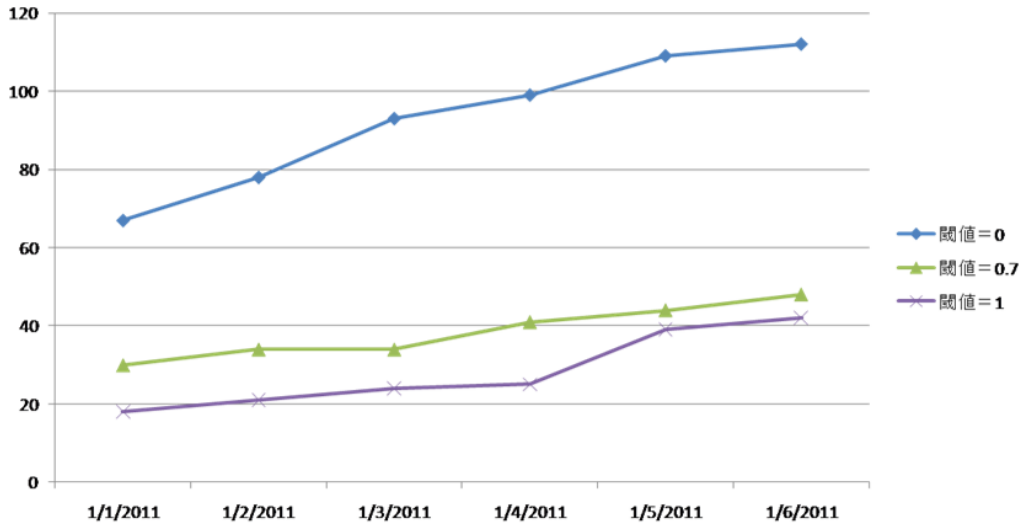


図 5.9: しきい値の考察

め，検知数が多い，管理者への負担を減らないことになる．

5.4 まとめ

本章では提案手法を検証した．実環境に応用し，警告を一部分消去した．ほかの警告を異常れべるによる管理者は判断しやすくなることと言える．

第6章　結論

本章では，本論文の成果をまとめ，本研究の目的を実現するために今後の展望を述べる．

6.1　まとめ

本研究の目的は，管理者の負担が少なくなるように侵入検知システムのログを調整することである．警告イベントを監視し異常な検知数かどうかを過去の傾向から判断する侵入検知システムのログ分析手法を提案した．

具体的に提案手法では警告イベントの検知数のばらつきを分析し，異常な検知数と正常な検知数に分類することである．過去期間において集約期間の警告イベントをシグネチャ名にクラスタリングし，シグネチャごとに標準偏差と平均を計算する結果を用いるとともに，3シグマ規則を適用して過去の検知数の分布を取る可能がある．そこで，検知数が異常かどうか判断できた．アノマリスコアとアノマリラベルを設定することにより，警告イベントの順番を決め，異常な警告イベントを優先同高に設定し，正常な警告イベントを優先度低で管理者に見せる．一部分の正常な警告イベントを除去するために，しきい値を設定した．実際のネットワーク環境と侵入検知システムの設定により，異なる結果があるため，過去期間と集約期間としきい値を管理者は自分で変化する可能がある．

実験環境に提案手法を適用して，毎日の警告イベントの中に正常な検知数（過去の検知数の濃い分布範囲にある検知数）が多いと確認できた．異常な警告と正常な警告を判断できた．その上で，異常な検知数を発見することと注意しべき警告（正検知）の関係，正常な検知数を発見することと無視しても問題ない警告（誤検知－フォールスポジティブ）の関係を検証した．

また，シグネチャ名にクラスタリングし，集約期間ごとに検出した警告イベントを管理者へ通知することはネットワーク管理者の負担を低減することに寄与できると考える．

6.2　今後の展望

提案手法による各パラメータ（過去期間，しきい値など）を設定し，警告イベントの中に異常検知数が判断できた．ただし，実際のネットワーク環境と侵入検知システムによって，異常な警告の中に本当の攻撃の割合，注意すべき警告の割合などが異なる．だから，研究室のネットワーク環境のほかに，いろいろな実際の環境に提案手法を適用しなけれ

ば，パラメータの設定方法を精密化する不可能がある．今後，もっとさまざまな組織で継続的な検証を必要とする．

また，集約期間 t ごとにクラスタリングした警告を管理者に見せる手法を利用しているため，提案手法を自動的なリアルタイムシステムに適用できる可能性がある．これから，そのシステムを作成し，すべて警告イベントログ分析結果だけをデータベースに蓄えて，リアルタイムで管理できると考える．

謝辞

本論文の作成にあたり，ご指導頂いた慶應義塾大学環境情報学部学部長村井 純博士，同学部教授 徳田 英幸博士，同学部教授 中村 修博士，同学部准教授 楠本 博之博士，同学部准教授 高汐 一紀博士，同学部准教授 三次 仁博士，同学部准教授 植原 啓介博士，同学部専任講師 重近範行博士，同学部専任講師 中澤 仁博士，同学部専任講師 Rodney D.Van Meter III 博士，同学部教授 武田 圭史博士に感謝致します．特に武田圭史博士は，常に私の研究について的確な助言をして下さりました．常に新しいアイディアと研究手法で私を道いていただき，何回も私に新しい視点や手本を見せていただきました．本当にありがとうございました．

そして，本研究を進めていく上で，様々な励ましと助言，お手伝いをいただきました，慶應義塾大学大学院政策メディア研究科博士研究員 水谷 正慶博士，同学部上原 雄貴氏，同学部 重松 邦彦氏に感謝致します．特に重松 邦彦氏は，親身に相談に乗っていただき，研究の方向性を指導や実装の細やかなケアをはじめとするあらゆる面で面倒を見ていただきました．氏なしでは卒論執筆だけでなく一年半に渡る充実した研究室生活が送れませんでした．本当に感謝致します．

研究に協力をしていただいた，梅田 昂翔氏，福岡 英哲氏，碓井 利宜氏，山本 智典氏，相見 眞男氏，藤原 龍氏，吉原 洋樹氏，由井 卓哉氏，Pham Van Hung 氏，Vu Xuan Duong 氏，中島 明日香女史，湯本 愛未女史，三ツ木 あかね女史と徳田・村井合同研究室の皆様に感謝致します．

研究室で苦楽を共にした Do Thi Thuy Van 氏，Nguyen Hung Long 氏，村上 滋希氏に感謝致します．彼らと一緒に研究をすることでお互いを刺激しあい，より質の高い議論や研究をすることができました．

最後に，今まであらゆる面で多大な助力を頂き，いつも私を支え励ましてくれた家族と友達に心から深謝と敬愛を表し，謝辞と致します．

以上を持って，謝辞といたします．

参考文献

[1] 独立行政法人情報処理推進機構 (IPA). http://www.ipa.go.jp/security/, 2010.

[2] Carnegie Mellon University's Computer Emergency Response Team (CERT). *Denial of Service Attacks*, 6 2001. http://www.cert.org/tech_tips/denial_of_service.html.

[3] Mozilla Developer Network. *JavaScript*, 2010. https://developer.mozilla.org/en/javascript.

[4] CIDF working group. Common Intrustion Detection Framework. http://gost.isi.edu/cidf/.

[5] 武田 圭史 and 磯崎 宏. **ネットワーク侵入検知**, pages 36–37. ソフトバンクパブリッシング株式会社, 6 2000.

[6] E.Rescorla. HTTP over TLS. *Internet RFC 2818*, 5 2000.

[7] Chad W. Engelgau and Sanjeet Singh. *Overview of SSL Acceleration Implementations*. Dell Power Solutions, 3 2002. http://www.dell.com/content/topics/global.aspx/power/en/ps1q02_ssl.

[8] 出口 雄一. *IDS* **と** *IPS*, 9 2006. http://itpro.nikkeibp.co.jp/article/COLUMN/20060830/246798/.

[9] 独立行政法人情報処理推進機構 (IPA). *Web Application Firewall* **読本**, 10 2010. http://www.ipa.go.jp/security/vuln/documents/waf.pdf.

[10] Microsoft セキュリティTechCenter. *SQL* **インジェクション攻撃とその対策**, 6 2008. http://technet.microsoft.com/ja-jp/library/dd362952.aspx.

[11] 上原 孝之. **情報セキュリティスペシャリスト**, pages 311–316. 株式会社 翔泳社, 10 2009.

[12] Tony Bradley. *Zero Day Exploits*. About.com Guide. http://netsecurity.about.com/od/newsandeditorial1/a/aazeroday.htm.

[13] Maciej Ogorkiewicz and Piotr Frej. *Analysis of Buffer Overflow Attacks*. Windows OS Security, 7 2004. http://www.windowsecurity.com/articles/Analysis_of_Buffer_Overflow_Attacks.html.

[14] Sourcefire Inc. Snort.

[15] 水谷 正慶. *NIDS*運用の効率化に関する研究, 2006.

[16] Frederic Cuppens and Alexandre Miege. Alert Correlation in a Cooperative Intrusion Detection Framework. *IEEE Symposium on Research in Security and Privacy*, 8 2004.

[17] Tadeusz Pietraszek. Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. In *In Proceedings of the Symposium on Recent Advances in Intrusion Detection (RAID)*, pages 102–124. Springer, 2004.

[18] 情報マネジメント用語事典. データマイニング. http://www.atmarkit.co.jp/aig/04biz/datamining.html.

[19] Klaus Julisch. *Using Root Cause Analysis to Handle Intrusion Detection Alarms*. PhD thesis, University of Dortmund, 2003.

[20] Stefanos Manganaris, Marvin Christensen, Dan Zerkle, and Keith Hermiz. A Data Mining Analysis of RTID Alarms. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 34:571–577, 2000.

[21] 藤田 直行. 誤検知低減の研究動向. In **電子情報通信学会論文誌**, 2006.

[22] Oracle Corporation. MySQL.