修 士 論 文

ネットワーク通信アラートを利用した攻撃予測に関する研究

指導教官　面　和成 准教授

審査委員主査　面　和成 准教授
審査委員　宮地 充子 教授
審査委員　平石 邦彦 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1210055 森 俊貴

提出年月: 2014 年 2 月

概 要

近年サイバー攻撃の数が増大している．特に，マルウェア等の自動プログラムを用いた攻撃が増加しており，その被害が深刻である．侵入検知システム（IDS）は，マルウェアに対する有効な対策の一つであると考えられている．しかし，実際は IDS は膨大なアラートを発生させることがあるため，管理者はこのアラートに対してどのように対処すればよいかの判断が難しい．これに対して，IDS のアラートを学習して攻撃を予測する手法（Nexat）が，Cipriano らによって 2011 年に提案された．この著者らは，カリフォルニア大学で開催されたハッキングコンペにおける攻撃ログを分析・評価し，Nexat の有効性を示した．しかしながら，この Nexat が IDS のアラートが必ずしも発生するとは限らないマルウェアの予測に使えるかどうかは不明である．

そこで，本研究では，まず Nexat がマルウェアの予測にどの程度有効であるかを評価した．IDS ではマルウェアダウンロードをアラートとして出ないため，マルウェアダウンロードを検知するためのローカルルールを作成し，学習に利用した．一方，予測に利用するアラートにはローカルルールを含ませない．このことにより，マルウェアに特化した学習に対し，通常の IDS アラートのみを用いた性能評価を実施した．この結果，およそ 7 割の確率でマルウェアダウンロードの予測を実現することができた．しかし，Nexat はネットワーク全般の攻撃予測をアラート単位で行うものであり，攻撃手法がある程度定例化されているマルウェアに対しては冗長であることが考えられる．

本研究ではさらに，Nexat における予測アルゴリズムをマルウェアに特化した簡易化を行った．これは，マルウェアがロボットプログラムであることを考慮して，主に予測フェーズを改良した．その結果，およそ 9 割の確率でマルウェアダウンロードの予測を実現することができた．

目 次

第1章　はじめに

1.1　研究背景

昨今において，ネットワークの外部からマルウェア感染などの攻撃行動による被害が増加している．マルウェア感染手段としては，Web ページからのダウンロードによるものや，不正アクセスを行ってからダウンロードを行うものがある．特に，最近ではボットネットによるマルウェア感染が深刻な問題となっている．そのため，マルウェア感染対策として，マルウェアダウンロードに関連する不正アクセスをいち早く検知し，マルウェアによる攻撃を止めることが考えられる．そこで，侵入検知システム（IDS）を活用した攻撃対策が考えられる．IDS は，ネットワークに流れるパケットを監視し，不正アクセスや攻撃の疑いがあるパケットを発見するとアラートとしてネットワーク管理者に通知し，当該通信記録をアラートログとして収集する．不正アクセスや攻撃命令の通信パターンはシグネチャ（ルール）としてあらかじめ定義することにより判別している．

しかし，IDS ルールは数が膨大であり，全てのルールを適用すると重要なアラートだけでなく重要でないアラートまでもが含まれ，膨大なアラートが発生してしまう．膨大なアラートが発生しているログへの対応はセキュリティ専門家による経験を基に判断する．ところがこのようなセキュリティ専門家は昨今不足しており，増加し続けているネットワークトラフィックや巧妙化・多様化する攻撃に対処し切れていないことが問題になっている．このため，IDS を導入したものの，ネットワーク管理者は，膨大に発生するアラートからマルウェアに関連するアラートを分析・予測することが困難になっている．

そこで，管理者によるアラート分析の負担を減らすために，各々のアラートに対してグループ化を行い，アラートの相関関係を自動的に抽出する研究が行われている [3, 4, 15, 16, 19]．アラートのグループ化を自動で行うことは，攻撃パターン抽出の自動化を行うことにつながり，管理者による攻撃分析が容易になるといった利点を持つ．また，これらの攻撃パターンを機械学習に用いると，学習した攻撃パターンを用いた新たな攻撃手法が生まれても，それらの攻撃を察知し，攻撃者が次に取る行動の予測が可能になる．Ciprian らによって提案された “Nexat”[2] では，過去に発生したアラートの相関関係を求め，それらを機械学習させることにより，リアルタイムで発生したアラートの次のアラートを予測している．この著者らは，カリフォルニア大学で行われたハッキングコンペにおいて発生した IDS アラートの分析・予測を行った結果，約 94%の確率で予測に成功したと主張している．しかし，Nexat ではマルウェアに対する有効性は示されていない．

本研究では，IDS アラートをベースとした攻撃予測手法 Nexat について，マルウェア研

究用データセットの一つである CCC DATAset 2011 を用いて有効性を評価する．

1.2 研究目的と成果

本研究ではまず，IDS アラートをベースとした攻撃予測手法 Nexat について，マルウェア予測にどの程度有効であるか，マルウェア研究用データセットの一つである CCC DATAset 2011 を用いて評価・考察する予備実験を行った．予備実験では，マルウェアダウンロード予測確率の推移，マルウェアダウンロード予測成功の平均確率，誤検知率で評価・考察する．予備実験を行うにあたり，Nexat アルゴリズム，マルウェアダウンロードアラート抽出プログラム，評価実験用プログラムは Python でフルスクラッチで実装した．その結果，およそ 70 %の確率でマルウェアを予測することができた．

一方，マルウェアによる攻撃はロボットプログラムによるものが多いことに着目し，本研究では Nexat アルゴリズムにおける予測フェーズの改良・簡単化を行い，予備実験と同様の性能評価を行った．その結果ほぼ 100 %の確率でマルウェアダウンロードの予測が実現され，Nexat と比べ予測精度の向上を図ることができた．

1.3 論文の構成

本稿の構成を示す．まず 2 章ではマルウェア対策の現状について説明する．次に 3 章では IDS アラートをベースとした攻撃予測やマルウェア検知の関連研究を紹介する．Nexat についても 3 章で詳しく説明する．4 章では性能評価に必要な評価手法やデータセットについて説明する．5 章ではまず予備実験として，Nexat を利用したマルウェア予測を行う．そして 6 章で Nexat を基に，マルウェア予測に特化した改良手法について検討する 7 章では Nexat，改良手法それぞれに対しマルウェア予測実験を行う．そして，8 章で予測実験の結果を基に未知マルウェアに対する予測性能やマルウェア侵入パターン，誤検知について考察する．最後に，9 章で本研究のまとめとして締めくくる．

第2章　マルウェア対策の現状

2.1　マルウェアの定義

マルウェア（malware）とは，ユーザのシステムの機密性，完全性，可用性を損なう目的や，ユーザを混乱させる目的で挿入される悪意のあるソフトウェア（malicious software）や悪意のあるコード（malicious code）の総称である．マルウェアはほとんどのシステムにおいて最も重大かつ深刻な問題を引き起こす外敵脅威である．種類として，主に以下のように分類される．

- ウィルス：ホストプログラムやデータファイルに忍び込ませることにより自己複製を行うプログラム．ウィルスは感染ユーザの操作を通じて起動することが多い．
- ワーム：自己複製もしくは自己完結型プログラムで，通常はユーザの関与なしで実行される．
- トロイの木馬：見かけ上は良性を装いながら，実際は悪意のある動作を行うためにユーザのシステムに侵入するプログラム．自己完結型でありプログラム自身の複製を行わないため，感染機能は持たない．
- 悪意のモバイルコード：悪意のある目的をもとに，通常はユーザによる明示的な指示なしにリモートシステムからローカルシステムに送信され，ローカルシステム内で実行されるソフトウェア．
- 攻撃ツール：マルウェア感染やシステム侵害を行うために利用される攻撃ツールもマルウェアの一つである．以下のものがよく知られている．
  - バックドア：特定の TCP または UDP ポートでコマンドを傍受するプログラム．ほとんどのバックドアは感染者のパスワードの取得や任意のコマンドを実行などを攻撃者のシステム上で行うことができる．
  - キーストロークロガー：キーボードの使用を監視する．
  - ルートキット：侵入を行った後，攻撃者がシステムの標準機能を悪意を持って改ざんするファイルの集合．侵入を隠蔽するためのログ改ざんツールや攻撃者の再侵入を容易に行うためのバックドアツールなどの攻撃などがパッケージ化されている．ルートキットによってシステムの改変を多く行うため，被害者は

ルートキットによって変更された箇所やルートキットの存在そのものを判断することが容易でない．

  - Web ブラウザプラグイン：本来は Web ブラウザを通じて特定の種類のコンテンツを表示するために利用するものだが，攻撃者はブラウザのあらゆる使用を監視するスパイウェアとして作成し，利用する場合もある．

- 混合攻撃：複数の感染手段や伝送手段を用いた攻撃．

攻撃者の多くは上記のマルウェアを複数利用することにより，効果的・継続的に攻撃を行っている．

攻撃者は目的に応じた攻撃を容易に行うために，攻撃に便利なユーティリティツールやスクリプトを収めた攻撃ツールキットを用いることが多い．攻撃ツールキットを攻撃目標のシステムにインストールすると，システム内のセキュリティをさらに侵害するのみならず，インストールされたシステムとは別のシステムに対し攻撃を行うこともできる．攻撃ツールキットに見られる典型的なプログラムとして以下のものがある．

- パケットスニファ：ネットワークトラヒックを監視し，パケットを捕捉するプログラム．パケットを捕捉することにより通信セッションを再構築し，通信の解析を行うことができる・

- ポートスキャナ：システム上のどのポートが開いているか（すなわち，通信可能であるか）を調べるためのプロトコル．代表的なものとして nmap[17] がある．攻撃者はポートスキャナを利用して攻撃目標システムの開放ポートのうち，脆弱性のあるポートがあるかどうかを調べることができる．

- 脆弱性スキャナ：ローカルシステムまたはリモートシステムの脆弱性を探し出すプログラム．

- パスワードクラッカ：OS やアプリケーションのパスワードを割り出す（クラッキング）プログラムで，パスワードの推測や可能性のあるすべてのパスワードに対する総当り攻撃（ブルートフォース攻撃とも呼ばれる）を行う．

- リモートログインプログラム：攻撃者が攻撃目標システムの制御や通信に利用する SSH や telnet などがそれに当たる．

上記のプログラムの多くは善意の目的で利用されているものである．例えば，パケットスニファはネットワーク管理者がネットワーク通信の問題を解決するのみならず，IDS を利用した攻撃検知はパケットをもとにした検知を行うために必要なため，セキュリティ管理のうえで重要なツールの一つである．このため，これらのプログラムが 1 つでもシステムに存在し，動作が行われたと言っても，必ずしも悪意のある行動が行われているとは限らないのである．

攻撃ツールキットの利用に関しても同様であることが考えられる．攻撃ツールの代表的なものとして Metasploit[11] がある．Metasploit は脆弱性スキャンニングでなく，既知の脆弱性を利用した様々な攻撃を選択肢から選ぶことにより簡単に実行することができる．しかし，Metasploit は本来セキュリティ管理者がシステムに脆弱性がないか確かめるペネトレーションテストで利用するツールキットである．このことから，攻撃ツールキットの利用に関しても必ずしも悪意のある行動とは限らないのである．

以上のように，攻撃ツールや攻撃ツールキットは善意の目的で利用されていることが多く，実際多くのツールが Web などを通じて容易に入手できる．しかし，攻撃ツールキットが容易に入手できることは悪意のある攻撃者が攻撃を簡単に行えることも意味する．さらに，攻撃者が攻撃ツールキットにあるマルウェアを改変して亜種を作成することも容易であることから，マルウェアのパターンマッチングによる検知が年々難しくなっていることが現状である．

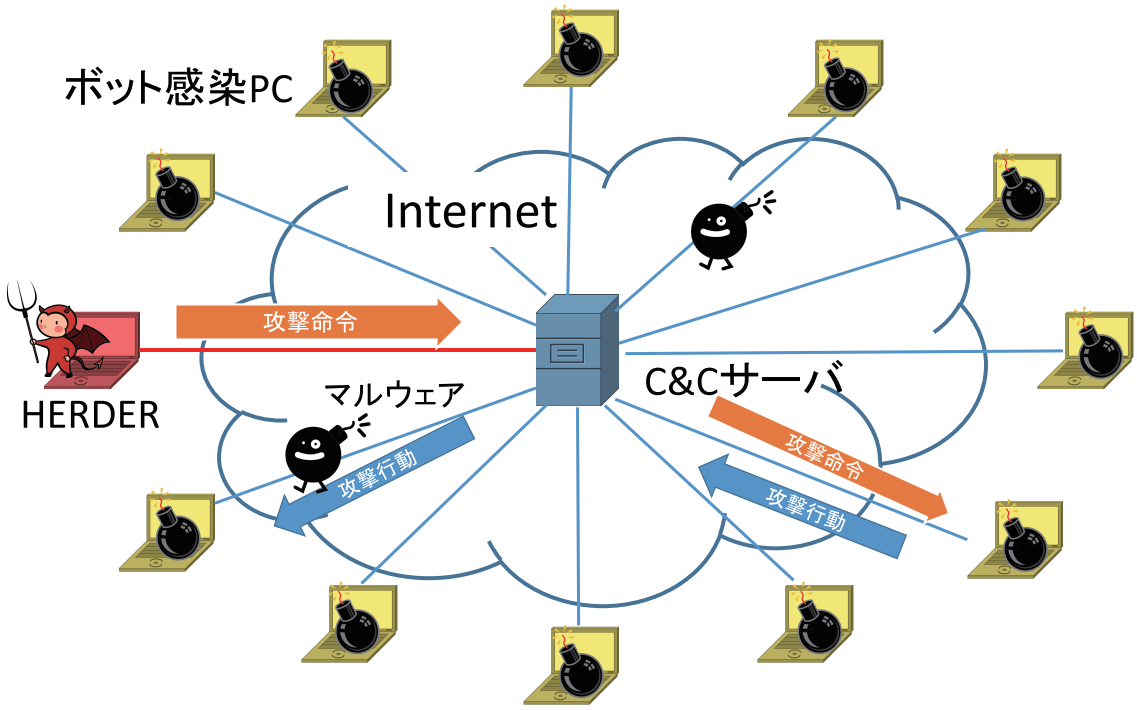

図 2.1: IRC ボットネットのイメージ

2.2 ボットネット

ボットネットとは，ボットと呼ばれるウィルスの一種が構成するネットワークの総称である [1]．通常のウィルスやワームと異なり，HERDER や MASTER と呼ばれる攻撃者の

指示により，DDoS 攻撃やスパムメールの送信のみならず，マルウェアの拡散にも利用されている．

ボットネットの多くは IRC プロトコルを利用することによりネットワークを構築している．図 2.1 に IRC プロトコルを用いたボットネットのイメージを示す．攻撃者は C&C（Command and Control）サーバを利用してボットに感染しているユーザの操作や指示を行っている．また，攻撃者は複数の CC サーバを用意してダイナミック DNS を利用したサーバ切り替えを行うことにより，ボットネットの堅牢性を確保している場合もある．なお，IRC の他に P2P や HTTP などのプロトコルを利用するボットネットも存在する．

ボットには DoS（サービス不能）攻撃をはじめとした攻撃機能，スパム送信機能，情報収集機能や感染機能などが実装されていることから，マルウェアの一つであると言うことができる．しかし，ウィルスやワームなどと言った他のマルウェアと大きく異なる点は，ボットに感染したノードのみのネットワークを構築する点である．さらに，攻撃者は CC サーバを通じてボット感染ノードの遠隔操作を行うことから，ボット感染ノードを踏み台にした攻撃を一斉に行うこともできる．

このため，ボットネットによるマルウェア攻撃の影響は計り知れないものとなっており，マルウェア対策においてもその背景を考慮しなければならない．

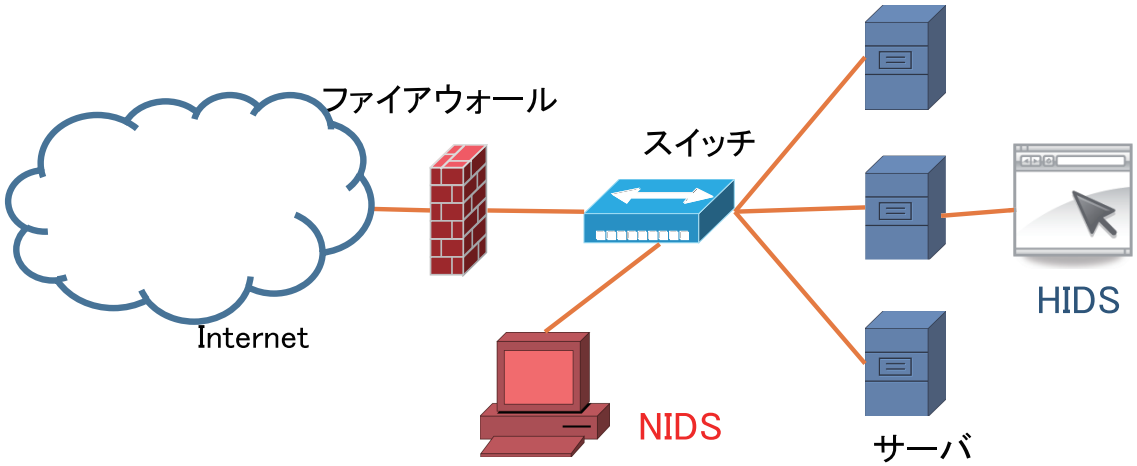

図 2.2: IDS の設置イメージ

2.3 IDS

Intrusion Detection System（IDS）とは，ネットワークやコンピュータ上での不正アクセスや攻撃命令などといった脅威をアラートとして管理者に知らせるシステムである．

不正アクセスや攻撃の検知手法は，ほとんどの IDS が「シグネチャベース」と呼ばれる手法を採用している．これは，不正アクセスや攻撃命令の通信パターンをシグネチャ

（ルール）としてあらかじめ定義し，設置箇所のネットワークトラヒックやファイルシステムなどを監視する．ほかにも,「アノマリベース」と呼ばれる手法があり，これは正常な状態と異常な状態のシステム動作を定義することにより侵入検知を行っている．

IDS の設置形態は以下の 2 つに大別できる（図 2.2）．

- ネットワーク型 IDS（NIDS）ネットワークに流入する全パケットを監視する．ネットワークのゲートウェイに設置することにより，ネットワークの配下にあるコンピュータに流入する全パケットを監視することができる．

- ホスト型 IDS（HIDS）サーバなどのコンピュータにソフトウェアとして組み込み，内部の通信やファイル構成などを監視する．コンピュータ内部で発生している攻撃行動を検知することができる．

標的型攻撃などと言った，企業などの組織に対する攻撃の検知にはネットワーク型 IDS が有効である．何故なら，社内ネットワーク（イントラネット）のほとんどはインターネットとパケットの送受信を行う際にゲートウェイに集約されるからである．

ネットワーク型 IDS の一つとして Snort[18] がある．Snort はオープンソースによるシグニチャベースのネットワーク型 IDS である．ルールと呼ばれるシグネチャの条件がネットワークに流入するパケットとマッチした場合，アラートとしてログファイル（アラートログ）に記録する．Snort のルールは.rule 形式のルールファイルとして保存されている．ネットワーク管理者はネットワークの実態に応じて Snort の設定やルールファールの改変を行うことにより，より的確な攻撃検知に努めている．

最新の Snort で動作する最新のルールには最新の攻撃手法を検知することができる．しかし，攻撃手法は日に日に巧妙化しているのが現状としてある．その結果，Snort ルールの数が莫大なものとなり，発生するアラートもルール数に比例して多く発生することから，ネットワーク管理者によるアラートベースの攻撃分析・予測が困難になりつつあることが問題である．

2.4 マルウェア解析

マルウェアの解析手法は以下の 2 つに大別できる．

- 動的解析：マルウェアを隔離環境下で実行させ，感染活動などの挙動を観察する．

- 静的解析：マルウェアのコードを解析することにより，マルウェアの機能や挙動を把握する．

動的解析では実際にマルウェアを動作させることから，マルウェアがもたらす脅威をその場で直感的に理解することができると言える．コンピュータ上でマルウェアを実行させる場合，感染対象コンピュータの復元作業にコストがかかることから，実際は仮想環境下で行われることが多い．しかし，マルウェアの機能によっては条件によって起動する機能や

ネットワークからの指示により起動する機能など，隔離環境下での解析では発現しないものもある．さらには仮想環境で動作されていることを検知することにより，動的解析を妨害しているマルウェアもあることから，動的解析のみではマルウェアの全容が解明できないことが問題である．

一方で静的解析では，マルウェアを実行させることなく逆アセンブルすることにより解析を行う．マルウェアのプログラムコードに着目することで，マルウェアの潜在的な機能や他のマルウェアとの類似性を考慮した分析が行える．しかし，プログラムコードによるマルウェアの解析は時間的コストが大きく，解析技術者のコードに関する知識や技術，経験に大きく依存する．さらに，昨今のマルウェアの多くは難読化・暗号化ツールを施すことにより静的解析に対する耐性を高め，解析技術者によるコード解析をより困難なものにしている．

第3章　関連研究

3.1　概観

IDS アラートを対象とした攻撃予測システムの提案は多く行わてきた．特に，アラートの前兆や関連アラートを発見するために，過去に発生したアラートのグループ化等を行うことにより，アラートの相関関係を見つけるものが多い [3, 4, 15, 16]．これらは，アラートのグループ化を自動で行うことにより，攻撃パターンの抽出を自動で行い，管理者による攻撃者の行動分析が容易に行えることを目指している．さらに，[6] では攻撃者の攻撃行動を可変長マルコフモデル（VLMM）で表現することにより，攻撃者の次の行動に関して確立過程を用いた予測を行っている．しかし，これらの研究におけるアラートの相関導出は事前に定義された攻撃の条件や攻撃結果に依存しているため，効果は限定的である．このことは，未知の攻撃や未知の攻撃の関係に対して，有効なアラートの相関を導出できないことを意味している．

一方，プラン認識（plan recognition）による攻撃者の行動予測も提案されている [5]．プラン認識とは，相手の行動を認識するための自然言語処理手法である．プラン認識を IDS に活用することにより，攻撃者の行動計画を推測し，攻撃者が次に取りうる一手について予測することを目的としている．しかし，この手法においても攻撃予測を行うためにはセキュリティ専門家の手を借りる必要があるため，この方式が広く使われることは少ないと言われている．

そこで，2011 年に Cipriano らによって提案された Nexat[2] では，攻撃者の過去に行った行動の相関関係を求め，リアルタイムで発生したアラートの次に発生したアラートの予測を行っている．Nexat ではアラートの相関関係を送信元・宛先 IP アドレス，送信元・宛先ポート番号，アラート発生時間のみを用いて完全自動で導出している．さらに，それらを機械学習させることにより，リアルタイムで発生したアラートの次に発生するアラートの予測を行っている．Cipriano らは，カリフォルニア大学において行われたハッキングコンペにおいて，およそ 94 %の確率で攻撃予測に成功したと主張している．

また，IDS の一つである Snort[18] のアラートを利用して，BOT 攻撃からマルウェアに感染するまでの通信を分析し，分析結果によりマルウェア検知に最適な IDS ルールの組み合わせを抽出する研究もなされている [19]．この研究では，マルウェア研究用データセットの一つである CCC DATAset 2011[7] の攻撃通信データを Snort に適用することにより，マルウェア検知ツールの IP アドレスとポート番号を用いて攻撃からマルウェア感染までに利用された IDS ルールの抽出を行っている．そして，抽出した IDS ルールを組み合わせ

ることにより，攻撃 BOT 通信の検知に最適な検知モデルを構築している．すなわち，マルウェア検知モデルを構築することにより，IDS アラートを利用して攻撃 BOT 通信を検知できることが述べられている．

一方，パケット数などといったネットワークのパケット特徴量に着目し，マルウェア感染トラヒックとマルウェアに感染していない正常トラヒックとの識別を行う研究もある [8]．これは，パケットのヘッダ情報から様々な特徴量を抽出し，マルウェア感染検知に有効な特徴量について検討している．この研究では，特徴量全体として，識別率が年々低下している傾向にあるという報告がされており，アノマリベースのマルウェア検知が年々困難になっていることを示唆している．

3.2 Nexat

Nexat[2] とは，Cipriano らにより提案された，IDS アラートを利用した攻撃予測手法である．これは，過去に発生した IDS アラートに対して機械学習を用いることにより，特定の IDS アラートに対する関連アラートを抽出し，リアルタイムで発生したアラートに対し，これから発生するアラートを予測する．すなわち，攻撃者の行動履歴から攻撃者の次の行動を予測する手法である．

この提案方式の動作は，データ抽出フェーズ，学習フェーズ，攻撃予測フェーズの 3 つに分けられている（図 3.1）．本章では，これら 3 つの動作について [2] の例を用いて解説する．

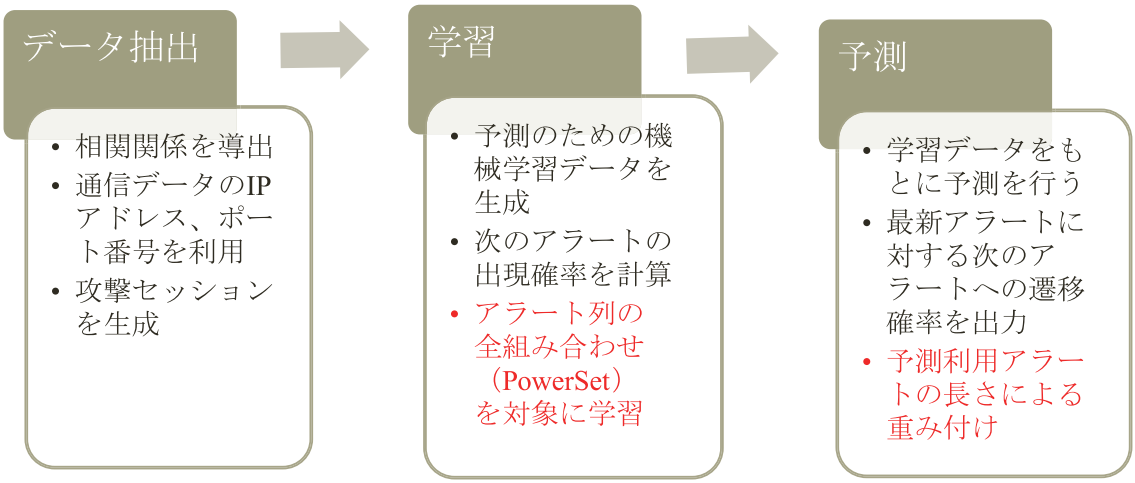

図 3.1: Nexat の動作フェーズ

3.2.1 データ抽出フェーズ

ここでは，IDS によって収集されたアラート情報を用いて，あるアラートに対する相関を導出する．アラートリストには，アラート名のほか，宛先 IP アドレス，送信元 IP アドレス，発生時刻などが明記されている．このアラートリストに対し，あるアラートを基点とした過去のアラートで相関関係を持つアラート群（攻撃セッションと呼ぶ）を求める．攻撃セッションの定義は以下の通りである．

Definition 1 (攻撃セッション) 攻撃セッションとは，以下の条件を満たすアラート列のことである．また次のうち少なくとも一つを満たせば，攻撃セッション S にアラート A を追加する：

- アラート A の宛先が攻撃セッション S 内にある過去のアラートの宛先と一致
- アラート A の送信元が攻撃セッション S 内にある過去のアラートの送信元と一致
- アラート A の送信元が攻撃セッション S 内にある過去のアラートの宛先と一致

さらに，アラート A が攻撃セッション S の最後のアラートから w 秒の時間ウィンドウ内に存在すべきである．

なお，攻撃セッションは次に述べる学習フェーズで利用する．

【データ抽出フェーズ】

1. アラートリストを読み込む
2. 新しいアラートから順に各アラートに対する攻撃セッションを導出する
3. 攻撃セッションリストを出力する

動作例として，表 3.1 のアラートリスト AL から攻撃セッションを求めるシナリオを図 3.2 を用いて説明する．まず，最後に発生したアラート E（Time=10）に着目し，このアラートの攻撃セッションを調べる．アラート E から過去のアラートをたどると，アラート D の宛先 IP アドレスがアラート E の送信元 IP アドレスと一致する．このことから，アラート D が発生した後にアラート E が橋渡しのように発生することが考えられる．アラート B に関しても，このアラートが発生してからアラート D，E が発生することが考えられる．また，アラート A の宛先 IP アドレスは，アラート E の宛先 IP アドレスと一致することから，アラート E と相関関係のあるアラートと考えられる．このように，アラート E の攻撃セッションは，アラート A，B，D であることがわかった．

以上の動作をアラートリスト AL 内の全アラートに対し適用し生成した攻撃セッションリスト H を表 3.2 にまとめる．

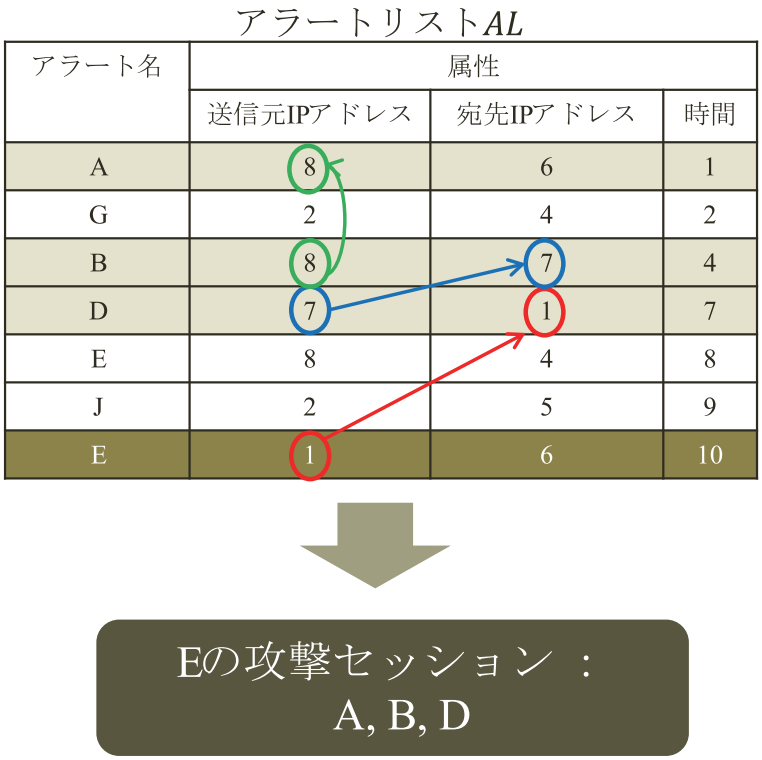

図 3.2: 攻撃セッション生成例

表 3.1: アラートリスト *AL* の例

アラート名	属性		
	送信元 IP アドレス	宛先 IP アドレス	時間
A	8	6	1
G	2	4	2
B	8	7	4
D	7	1	7
E	8	4	8
J	2	5	9
E	1	6	10

表 3.2: 攻撃セッションリスト H の例

アラート名	攻撃セッション
E	{A,B,D},{A,B,G}
J	{G}
D	{A,B}
B	{A}
G	ϕ
A	ϕ

3.2.2 学習フェーズ

前節で導出したアラート相関関係を用いて，アラートの発生確率を求める．まず，あるアラートに対し相関するアラート全ての組合せを考える．例えば，アラート E と相関するアラートが $H[E] = \{\{A, B, D\}, \{A, B, G\}\}$ であった場合，E は $\{A, B, D, G\}$ のいずれかのアラートと相関することが考えられる．ここから，E と相関することが考えられるアラートの組合せ（Power set）は

$$PowerSet(H[E]) = \{\{A\}, \{B\}, \{D\}, \{G\}, \{A, B\}, \{A, D\}, \cdots, \{A, B, D, G\}\} \quad (3.1)$$

のいずれかである．この中のすべてのアラート集合において，E と相関しているアラートそれぞれの相関回数を計算し，アラート関連回数の合計を母数に取って正規化を行う．例えば，アラート B が発生した後，0.67 の確率でアラート E が発生すると定義できる．これにより，過去のアラートに対し，次に発生するアラートの確率を表すことができる．これを学習データとする．学習データの一例を表 3.3，3.4 に示す．

【学習フェーズ】

1. 攻撃セッションリストを読み込む
2. 攻撃セッションの PowerSet を導出する
3. PowerSet で生成されたアラート列が攻撃セッションリスト内で出現した回数をアラート遷移回数として計算する
4. アラート列に対するアラート遷移回数の合計を母数に，遷移確率を求める
5. 学習データを出力する

3.2.3 攻撃予測フェーズ

前述したデータ抽出フェーズ，学習フェーズに関しては過去に発生したアラートの群を利用した．このフェーズでは，学習データを用いて現時点で発生しているアラート（ライブストリームアラートと呼ぶ）を利用し，次に発生するアラートを予測する．

表 3.3: 学習データの例（正規化前）

{A}	{1,B},{1,D},{2,E}
{B}	{1,D},{2,E}
{D}	{1,E}
{G}	{1,E},{1,J}
{A,B}	{1,D},{2,E}
⋮	⋮

表 3.4: 学習データの例（正規化後）

{A}	{0.25,B},{0.25,D},{0.5,E}
{B}	{0.33,D},{0.67,E}
{D}	{1,E}
{G}	{0.5,E},{0.5,J}
{A,B}	{0.33,D},{0.67,E}
⋮	⋮

表 3.5: 予測結果リストの例

アラート名	{B}	{D}	{B,D}	Weighted sum	Predicted probability
D	(0.33,1)	(0,1)	(0,2)	0.33	0.083
E	(0.67,1)	(1,1)	(1,2)	3.67	0.917

まず，ライブストリームアラートに対し，すでに導出している学習データと照合する．例として，収集したライブストリームアラートを $S_I = \{B, D\}$ とすると，これらの Power set は

$$PowerSet(S_I) = \{\{B\}, \{D\}, \{B, D\}\} \tag{3.2}$$

となる．この中のすべてのアラート集合において，学習データから，次に発生するアラートの確率を呼び出し，予測結果リストに予測利用アラートの長さとともに追加する．これは，予測に利用するアラート数が多いほど，アラートを予測する価値があることによる．このアラート長と確率をそれぞれの予測利用アラートで掛け合わせることにより，$PowerSet(S_I)$ と関連するアラートの重み（Weighted sum）を計算することができる．そして，$PowerSet(S_I)$ と関連するアラートそれぞれの重みの割合により，次に発生するアラートの予測確率（Predicted probability）がわかる．予測結果の一例を表 3.5 に示す．

【予測フェーズ】

1. 学習データ，ライブストリームを読み込む．
2. ライブストリームの攻撃セッションを導出する．
3. 攻撃セッションの PowerSet を導出する．これを予測利用アラートとする．
4. 予測利用アラートから予測可能アラートへの遷移確率を学習データより呼び出す．
5. 予測利用アラート長×予測可能アラート遷移確率をアラートの重み（ Weighted sum）として記録
6. 4-5 を全予測利用アラートに対し実施．
7. 予測可能アラートすべてに対する重みの割合を計算し，これを予測確率（Predicted probability）とする．

3.3　マルコフモデルを利用した攻撃予測 [6]

IDS では，ネットワークやサーバへの脅威をトラヒックデータをもとにアラートとして出力している．攻撃者の攻撃行動は，複数の IDS アラートを順序付けることにより表現することができる．本研究では，IDS アラートに現れる攻撃者の攻撃行動を可変長マルコフモデル（VLMM）で表現し，攻撃者の次の行動に関して予測している．

IDS アラートに現れる攻撃者の攻撃行動を VLMM で表現するためには，まず攻撃者が発生させた IDS アラートの相関関係を導出しなければならない．IDS のアラート情報を XML 形式の情報にまとめる．

ここで，攻撃トラック（攻撃列）を $\Psi = \{\sigma_i, i = 1, 2, 3...\}$ とおく．ただし，σ_i は攻撃が発生した時刻を含む攻撃アラート列 $\{a_{(i,1)}, a_{(i,2)}, a_{(i,3)}...\}$ である．$a_{(i,j)}$ はアラート属性 $v_1, v_2, v_3...$ を含むため

$$\sigma_i = \{a_{(i,1)}, a_{(i,2)}, a_{(i,3)}...\} = \left\{ \begin{pmatrix} v_1 \\ V_2 \\ ... \end{pmatrix}_i, 1 \right\} \tag{3.3}$$

また，攻撃行動をマルコフモデルに適用するために，攻撃の一連の順序を接尾辞木（Suffix tree）で表現している．

次に，以下の行動を組み合わせた攻撃 +FGGFGF* について考える．+は攻撃の根であり，*が攻撃の終端を示す．

- F: “WEB-IIS nsiislog.dll access”
- G: “WEB-MISC Invalid HTTP Version String”

この攻撃行動についての Suffix tree を図 3.3 に示す．Suffix tree は固定順序モデルであるため，VLMM に結合することも可能である．

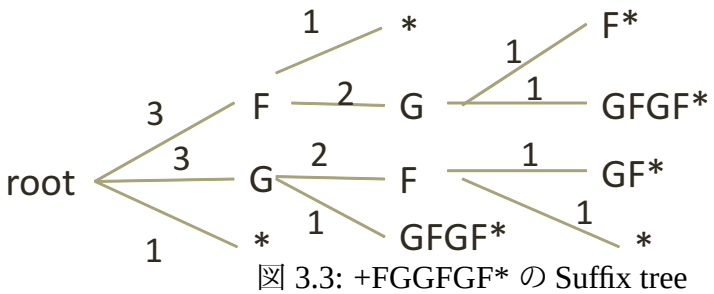

図 3.3: +FGGFGF* の Suffix tree

3.4 IDS アラートを利用したマルウェア攻撃分析

IDS で発生したアラートをもとにしたマルウェア攻撃分析として，マルウェア攻撃の通信データによって発生した IDS アラートを分析し，ボットによる攻撃からマルウェアに感染するまでに検知で利用された IDS ルールを抽出する研究がある [19]．具体的には，まずマルウェア研究用データセットの一つである CCC DATAset 2011 の攻撃通信データを Snort に適用し，アラートログを作成する．なお，マルウェアの検知のために “This program cannot by run in DOS mode” と “Windows Program” がパケット内部に存在した場合，アラートとして通知するローカルルールをそれぞれ定義している．そして，アラートログと攻撃元データから，マルウェアに感染した通信と感染に起因した攻撃のみを抽出している．

2010 年 8 月 18 日から 31 日までと，2011 年 1 月 18 日から 31 日までの攻撃元データでマルウェア攻撃通信の分析を行った結果，741 検体中 733 検体の検知に成功し，残り 8 検体は検知することができなかった．検知漏れの原因として，ハニーポット自らが BOT からマルウェアのダウンロードを試みたため，Snort では正常な通信として判断したためと述べられている．

図 3.4 にマルウェア分析実験から提案された感染ルール SID: 14415 への攻撃モデルを示す．SID: 14415 は “This program cannot by run in DOS mode” がパケット内に存在した場合のルールであり，この研究ではマルウェアダウンロードを指している．感染ルールを生成する際に行われた分析では，攻撃から感染までの組み合わせとして最も多かったものは SID: 2466 から SID: 14415 である．SID: 2466 は Windows のファイル共有の脆弱性を狙った攻撃である．これ以外のルールも「NETBIOS SMB」に関するルールがほとんどであることから，攻撃者は SID: 14415 の攻撃を行う前に Windows ファイル共有の脆弱性を狙うことがわかる．

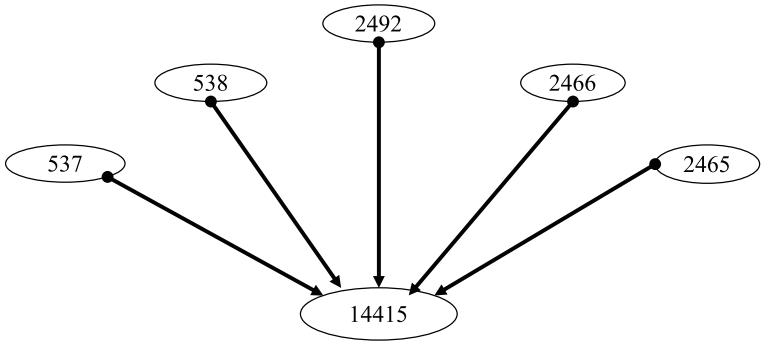

図 3.4: SID: 14415 への攻撃モデル

3.5　パケット特徴量に着目したマルウェア検知

一方，マルウェア感染検知に有効な特徴量についての考察も行われている．[8] ではパケット数やパケットサイズ，到着間隔などといったパケットの特徴量に着目し，マルウェアに感染しているトラヒックとマルウェアに感染していない正常なトラヒックを識別する実験により特徴量評価を行っている．性能評価に用いる感染トラヒックとして CCC DATAset 2009，2010，2011[7] の攻撃通信データから，マルウェア感染以降のトラヒックのみを利用している．正常トラヒックには CCC DATAset と同じデータ収集日にあるイントラネットにおけるトラヒックデータを利用している．

性能評価では，まず特徴量全体の傾向として，年を追うごとに識別率が低下していることが報告されている．原因として，年々移り変わるマルウェアや新種のアプリケーションの発生に伴う，トラヒックの複雑化や多様化があると推測している．

第4章　準備

4.1　攻撃セッション

攻撃セッションとは，一つの攻撃シナリオにおいて，目的とする攻撃に関連する IDS アラート列のことである．本研究では，Nexat[2] で定義された次の攻撃セッションを利用する．

Definition 2 (攻撃セッション [2]) 攻撃セッションとは，以下の条件を満たすアラート列のことである．また次のうち少なくとも一つを満たせば，攻撃セッション S にアラート A を追加する：

- アラート A の宛先が攻撃セッション S 内にある過去のアラートの宛先と一致
- アラート A の送信元が攻撃セッション S 内にある過去のアラートの送信元と一致
- アラート A の送信元が攻撃セッション S 内にある過去のアラートの宛先と一致

さらに，アラート A が攻撃セッション S の最後のアラートから w 秒の時間ウィンドウ内に存在すべきである．

ここで，攻撃者の目的の攻撃を O としたとき，攻撃セッションは $S = A_1, A_2, \ldots, A_m, O$ となる．つまり，アラート列 $A_1, A_2, \ldots, A_m$ は O の前兆となる関連アラート列と捉えることができる．もちろん，O が O の関連アラートになる場合もある．攻撃者の目的の攻撃 O をマルウェアダウンロード M と想定すると，マルウェアダウンロードの前兆となるアラートは $M = A_1, A_2, \ldots, A_m, O$ と捉えることができる．

4.2　誤検知

マルウェアダウンロード予測における正しい検知は以下の 2 つが考えられる．

(a) 予測確率が高い値において，マルウェアダウンロードが発生した (True Positive)

(b) 予測確率が低い値において，マルウェアダウンロードが発生しなかった (True Negative)

表 4.1: マルウェア予測における結果パターン

	マルウェアダウンロードが発生した	マルウェアダウンロードが発生しなかった
予測確率が高い値	True Positive（正しい検知）	False Positive（誤検知）
予測確率が低い値	False Negative（誤検知）	True Negative（正しい検知）

なお，予測確率の高低は閾値をもとに判断する．

一方，マルウェアダウンロード予測における誤検知として次の 2 つが考えられる

(c) 予測確率が低い値にも関わらずマルウェアダウンロードが発生した (False Negative)

(d) 予測確率が高い値にも関わらずマルウェアダウンロードが発生しなかった (False Positive)

マルウェアダウンロード予測において，事前にマルウェアダウンロードを予測できなかった (c) の誤検知は致命的である．一方，(d) の誤検知は重要ではない．なぜなら，たとえマルウェアの侵入がなかったとしても，実際にはマルウェアの前兆に似た攻撃行動が行われた事実がアラートによって示されたからである．このことから，False Positive に関しては誤検知と呼べない内容であることが多いと考えられる．また，Snort がアラートを出力しない時はマルウェア予測確率はゼロあることも言える．

以上の結果パターンを表 4.1 にまとめる．

マルウェアダウンロード予測の信頼性に対する評価指標として，表 4.1 から以下の 4 つの指標を考えることができる．

- True Positive Rate：マルウェアダウンロードが発生した時において，直前の予測確率が高かった割合．
- True Negative Rate：マルウェアダウンロードが発生しなかった時において，直前の予測確率が低かった割合．
- False Positive Rate：マルウェアダウンロードが発生しなかった時において，直前の予測確率が高かった割合．
- False Negative Rate：マルウェアダウンロードが発生した時において，直前の予測確率が低かった割合．

このうち，誤検知率と呼べるものは False Positive Rate（FPR）と False Negative Rate（FNR）である．

False Negative Rate は以下の式で表される．

$$\mathrm{FalseNegativeRate} = \frac{\mathrm{num(FalseNegative)}}{\mathrm{num(TruePositive)} + \mathrm{num(FalseNegative)}} \tag{4.1}$$

(4.1) でわかるように，マルウェア解析における False Negative Rate とは，全ての予測対象マルウェアの中で (c) の誤検知が含まれる割合を示している．前述の通り，事前にマルウェアを予測できなかった (c) の誤検知は致命的であるため，False Negative Rate はできるだけ小さい値であることが望ましい．

False Positive Rate は以下の式で表される [14].

$$\mathrm{FalsePositiveRate} = \frac{\mathrm{num(FalsePositive)}}{\mathrm{num(TrueNegative)} + \mathrm{num(FalsePositive)}} \tag{4.2}$$

False Positive Rate はマルウェアダウンロードが発生しなかった時に発生した IDS アラート全ての中から，直前の予測確率が高い値が含まれる割合を示している．False Negative Rate の母数はマルウェアダウンロードが行われた回数であることに対し，False Positive Rate の母数はマルウェア予測確率の導出が行われたものの，マルウェアダウンロードが行わてなかった事象の合計であることに注意されたい．

4.3 CCC DATAset

4.3.1 概要

本研究では，MWS2011 の研究用データセット [7] のうち，サーバ型ハニーポットのデータセットである CCC DATAset2011 を用いた性能評価を行った．このデータセットには，感染 PC 群からの攻撃を一手に受けるハニーポットで収集したマルウェア検体のハッシュ値，ハニーポットがマルウェアを取得した日時を記憶したログ（攻撃元データ）と，ハニーポットのパケットキャプチャログ（攻撃通信データ）がある．図 4.1 に CCC DATAset 2011 の全体図を示す．

本研究では，このうち攻撃元データと攻撃通信データを用いてマルウェア関連アラートを抽出し，性能評価を行う．

4.3.2 攻撃元データ

攻撃元データはハニーポット内のウィルススキャナで出力されたログデータである．マルウェア検体の取得時刻ごとに，送信元・宛先 IP アドレス，送信元・宛先ポート番号，TCP/UDP，マルウェア検体の SHA1 ハッシュ値，マルウェア名，ファイル名が記載されている．CCC DATAset 2011 では 2010 年 5 月 1 日から 2011 年 1 月 31 日までの 9ヶ月間に 72 台のハニーポットにより収集した．なお，ウィルススキャナによって特定できなかったマルウェアはマルウェア名が “UNKNOWN” となる．今後，マルウェア名が “UNKNOWN” となるマルウェアを未知マルウェアと呼ぶ．

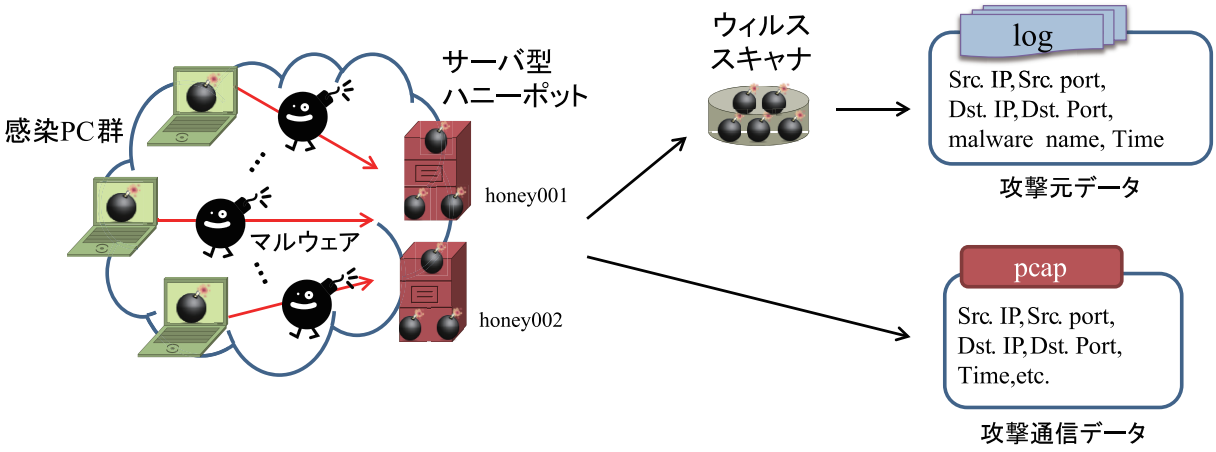

図 4.1: CCC DATAset 2011

4.3.3 攻撃通信データ

攻撃通信データとは，2 台のハニーポット（honey001，honey002）で収集されたパケットのキャプチャログである．収集期間は 2010 年 8 月 18 日～8 月 31 日，2011 年 1 月 18 日～1 月 31 日までである．このログにはマルウェアダウンロードに関係のない通信のパケットも含まれるため，このまま IDS に適用させるとマルウェアに関係のないアラートも発生する．一方，IDS では通常マルウェアダウンロードはアラートとして出力されない．そのため，マルウェア感染に着目したアラート分析を行うためには，ファイルダウンロードが行われた場合はアラートとして出力するローカルルールを作成し，ローカルルールを含む IDS アラートログと攻撃元データとの関連付けが必要である．具体的な関連付け手法については 5.1 章で述べる．

本研究では 2011 年 1 月 18 日から 1 月 24 日までの攻撃通信データをそれぞれ学習に利用し，2011 年 1 月 25 日から 1 月 31 日までの攻撃通信データによって出力された IDS アラートをリアルタイムアラートとして予測運用を行った．

第5章　Nexatを利用したマルウェア予測手法

Nexatでは過去に発生したIDSアラートの相関関係（攻撃セッション）をアラート送信元・宛先IPアドレスとアラート発生時刻を利用して求め，それらを学習・予測において利用している．IDSアラートは攻撃事例を示しているものであるため，マルウェアダウンロードという事象に対する攻撃セッションをNexatで導出すれば，マルウェアダウンロード攻撃の手法の分析が容易行えるだけでなく，マルウェアダウンロードを予測するための学習を実現することが容易に考えることができる．

Nexatでは，IDSがアラートを出力しない時は予測確率がゼロとなる点も特徴としてある．しかし，IDSでは通常マルウェアダウンロードはアラートとして出力されない．一方，ハニーポットで収集されるパケットキャプチャログではファイルダウンロードの事例も観測することができる．ところが，ダウンロードされたファイルがマルウェアかどうかの判断はパケットキャプチャログのみでは行うことができない．

このため，マルウェアダウンロードが行われるまでに攻撃者が行った行動をIDSベースで分析を行うためには，まずIDSでダウンロードを検知し，そのダウンロードされたファイルがマルウェアであるかどうかの判断が必要であると考える．

本研究では，Nexatのアルゴリズムをマルウェアに適用するために，IDSを利用してマルウェアダウンロードアラートの抽出を行った．次節で抽出手法について説明する．

5.1　前処理

本研究ではNexatと同様にフリーIDSツールの一つであるSnortを利用し，パケットキャプチャログからアラートを発生させた．Snortも含め，通常IDSではマルウェアダウンロードはアラートとして出力されない．このため，[19]を参考に，“This program cannnot by run in DOS mode” と “Windows Program” がパケット内部に存在した場合，それをアラートとして出力するようにローカルルールを追加した．ただし，上記のローカルルールにより出力されるアラート（ローカルアラート）にはマルウェアと関係ない誤検知のものも含まれる．このため，以下の手順により誤検知アラートを除外し，マルウェアダウンロードと関連するローカルアラートのみを抽出した．

【マルウェアダウンロードアラートの抽出】

1. 攻撃通信データに Snort を適用させ，ローカルアラートを含むアラートリストを出力する．
2. 攻撃元データを呼び出し，アラートリスト内のローカルアラートそれぞれに対し，攻撃元データの送信元 IP アドレス，送信元ポート番号，宛先 IP アドレス，宛先ポート番号の一致を確認する．
3. 攻撃元データとローカルアラートの時間差が 5 秒以内であればマルウェアダウンロードアラートとみなし，この時間差より大きいものは誤検知としてアラートリストから削除する．

マルウェアダウンロードアラート抽出動作を図 5.1 にまとめる．

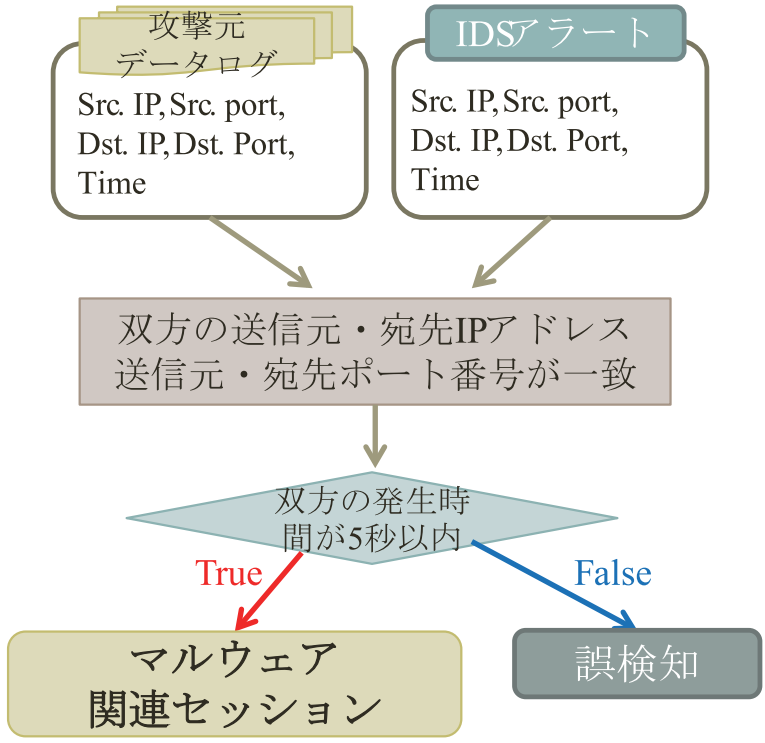

図 5.1: マルウェアダウンロードアラートの抽出

以上の操作によりマルウェア学習に特化した IDS アラートを出力し，これをマルウェア予測のための学習に利用する．

5.2 データ抽出フェーズ

前節ではマルウェア予測に特化した IDS アラートリストを出力した．これを Nexat におけるデータ抽出フェーズに適用すると，マルウェアに関する攻撃セッションが時系列で現れる．この攻撃セッションを機械学習に利用する．

表 5.1: 学習利用アラート

SID	アラート名	認知年月
14782	NETBIOS DCERPC NCACN-IP-TCP srvsvc NetrpPathCanonicalize ⋯	2008/09
7209	NETBIOS DCERPC NCACN-IP-TCP srvsvc NetrPathCanonicalize ⋯	2006/08
9422	SPECIFIC-THREATS msblast attempt	2003/07
9607	NETBIOS DCERPC NCACN-IP-TCP ISystemActivator ⋯	2003/09
9419	SPECIFIC-THREATS sasser attempt	2004/04
2508	NETBIOS DCERPC NCACN-IP-TCP lsass ⋯	2004/04
17344	SHELLCODE x86 OS agnostic xor dword decoder	
15306	FILE-IDENTIFY Portable Executable binary file magic ⋯	

【データ抽出フェーズ】

1. アラートリストを読み込む
2. 新しいアラートから順に各アラートに対する攻撃セッションを導出する
3. 攻撃セッションリストを出力する

5.3 学習フェーズ

学習フェーズはオフラインで事前に行えるものである．そのため，前節で述べている通り，マルウェアを特定するための学習に特化したアラートを用いた．データ抽出フェーズで出力された攻撃セッションから学習させる方法は Nexat[2] と同じである．表 5.1 の通常のアラートとマルウェアの侵入を検知するアラートとの相関関係を学習させることにより，攻撃者がマルウェアダウンロードを行うまでに実施される攻撃（マルウェア関連攻撃セッション）を集計し，マルウェア関連攻撃セッションからマルウェアダウンロードへの遷移を確率で導出することができる．

【学習フェーズ】

1. 攻撃セッションリストを読み込む
2. 攻撃セッションの PowerSet を導出する
3. PowerSet で生成されたアラート列が攻撃セッションリスト内で出現した回数をアラート遷移回数として計算する
4. アラート列に対するアラート遷移回数の合計を母数に，遷移確率を求める
5. 学習データを出力する

5.4　予測フェーズ

攻撃予測では，実時間で発生したIDSアラート（ライブストリーム）に対し，Nexatにおける予測フェーズを適用させる．なお，ライブストリームには前節で追加したローカルルールにより発生したマルウェアダウンロードアラートは含まない．このことは，通常のIDSアラートのみでマルウェアダウンロードの予測を意味する．

第6章　改良手法

6.1　マルウェア予測に特化した攻撃予測への検討

Nexat では学習と予測において，あるアラートの攻撃セッションにおけるすべての組み合わせ（PowerSet）を生成する動作を行う．これは，攻撃セッション内のアラートの順番が入れ替わったり，途中で抜けたりするようなアラート列に対しての予測ができるようにするためのものである．このように，マルウェアダウンロード予測を対象とした学習においても，PowerSet を利用することはロバストネスの面で効果的である．しかし，PowerSet を利用することによる弊害も考えられる．例えば，PowerSet により生成されたアラート列は，実際に発生していないが将来起こりうるアラート列をも含む．このような学習データに基づき，実際のアラート列に対して PowerSet を用いて予測確率を導出する Nexat は，人間が介在する攻撃に対しては効果的かもしれないが，ロボットプログラムで動作するマルウェアに対しては予測確率を低くする恐れが考えられる．

そこで本研究では，Nexat に対して予測フェーズを簡易化し，マルウェアを考慮した予測フェーズの改良を行った．データ抽出並びに学習では Nexat のアルゴリズムを流用するが，攻撃予測において PowerSet を用いて予測確率を導出する部分をやめ，学習データを直接的に活用して予測確率を算出する．

6.2　データ抽出フェーズ

データ抽出フェーズでは，IDS によって収集されたアラート情報を用いて，あるアラートに対する相関（攻撃セッション）を導出するフェーズである．このフェーズに関しては Nexat[2] と同一のアルゴリズムを利用する．また，攻撃セッションの定義についても Nexat と同一のものを用いる．

【データ抽出フェーズ】

1. アラートリストを読み込む
2. 新しいアラートから順に各アラートに対する攻撃セッションを導出する
3. 攻撃セッションリストを出力する

6.3 学習フェーズ

学習フェーズでは，前節で導出した攻撃セッションを用いてアラートの発生確率を求める．このフェーズにおいても Nexat と同じアルゴリズムを用いる．

【学習フェーズ】

1. 攻撃セッションリストを読み込む
2. 攻撃セッションの PowerSet を導出する
3. PowerSet で生成されたアラート列が攻撃セッションリスト内で出現した回数をアラート遷移回数として計算する
4. アラート列に対するアラート遷移回数の合計を母数に，遷移確率を求める
5. 学習データを出力する

6.4 予測フェーズ

前節で導出した学習データを利用し，実際のアラート（ライブストリーム）よりマルウェアダウンロード確率を導出するフェーズである．学習データにはマルウェアダウンロード以外の攻撃への遷移確率も導出されているが，今回の性能評価ではマルウェアダウンロードへの遷移確率のみ利用する．これをマルウェアダウンロード予測確率と呼ぶ．

Nexat[2] では予測フェーズにおいて，まず実際に発生したアラートを単位時間収集し，最新アラートと相関するアラート全ての組み合わせを PowerSet 関数を用いて生成していた．そして，PowerSet で生成されたアラート列それぞれに対し学習データより予測確率を呼び出し，さらにアラート列の長さを重みに，予測可能アラート全体における重みの割合をアラート予測確率としていた．

しかし，今回の予測はマルウェアダウンロードのみを対象としている．他のアラートの予測を行わず，過去に発生したマルウェアダウンロードの攻撃パターンをもとにした攻撃予測を行うと，Nexat を利用した場合，PowerSet を生成する動作が必ずしも効果的であるとは言えないことが考えられる．そこで，本研究では，あるアラートに対し相関するアラート全ての組み合わせを生成する動作を廃し，実際に発生したアラートをもとに生成した攻撃セッションをもとに，学習データからマルウェアダウンロードへの遷移確率を導出する．これをマルウェアダウンロード予測確率と呼ぶ．

【予測フェーズ】

1. 学習データ，ライブストリームを読み込む．
2. ライブストリームの攻撃セッションを導出する．
3. 導出した攻撃セッションからマルウェアダウンロードへの確率を算出して返す．

例えば，ライブストリームにより生成された最新アラートの攻撃セッションが $\{A, B\}$ であったとする．また，マルウェアダウンロードアラートを E と考える．表 3.4 の学習データより，$\{A, B\}$ からマルウェアダウンロード E までの遷移確率は 0.67 である．この確率が最新アラートにおけるマルウェアダウンロード予測確率である．

改良手法の予測フェーズで利用する攻撃セッションはライブストリームで流れているアラートから求めたものをそのまま利用する．すなわち，過去に発生した攻撃と，現在発生している攻撃とのパターンマッチングをもとに攻撃予測を行う．Nexat における PowerSet の生成と，予測アラート数による重み付け計算を行わないことから，予測計算の簡単化を実現したと言える．

提案方式の動作フェーズを図 6.1 にまとめる．

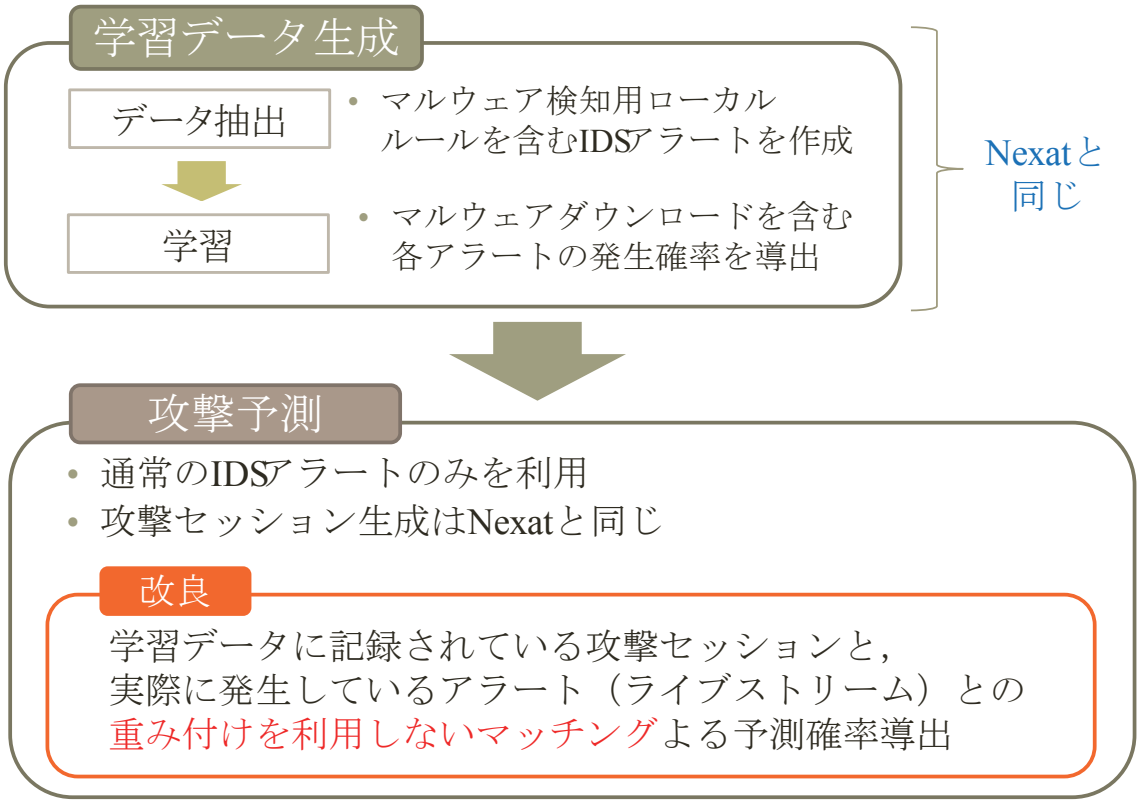

図 6.1: 提案方式の動作フェーズ

第7章　マルウェア予測実験

7.1　目的

本実験では，CCC DATAset に対して Nexat と改良手法による解析を行い，マルウェアの侵入を予測することを目的とする．

7.2　実験内容

本研究では CCC DATAset 2011[7] にある各種データのうち，2 台（honey001, honey002）の攻撃元データと攻撃通信データを用いて，改良手法の有効性を評価するためのマルウェア予測実験を行う．なお，Nexat アルゴリズム並びに改良手法は Python によりフルスクラッチで実装し，評価した．

本研究における学習期間は 2011 年 1 月 18 日から 1 月 24 日までである．この期間の攻撃通信データを Snort に適用し，それぞれ IDS アラートリストを生成する．次に，これらの IDS アラートリストそれぞれに対し攻撃セッションの作成・データ抽出並びに学習を行う．学習に用いたアラートは表 5.1 に示す．なお，ここでアラートに対する認知年月とは，Snort のアラートログに示されている脆弱性情報の参考 URL の一つである Microsoft Security TechCenter[12] において脆弱性情報が公開された年月を示している．学習は過去に発生したマルウェアダウンロードアラートを含む IDS アラートを収集し，オフラインで学習することを前提とする．

次に，出力された学習データに対し，2011 年 1 月 25 日から 1 月 31 日まで延べ 7 日間のマルウェア予測確率を時系列で導出する．予測運用に利用する IDS アラートにはマルウェアダウンロードアラートを含まないものを利用する．すなわち，通常の IDS アラートのみでマルウェアダウンロードの予測が行えるか評価する．

7.3　実験手順

7.3.1　データ抽出

前節ではマルウェア予測に特化した IDS アラートリストを出力した．これを改良手法におけるデータ抽出フェーズに適用すると，マルウェアに関する攻撃セッションが時系列で現れる．この攻撃セッションを機械学習に利用する．

7.3.2 学習

Nexat，改良手法それぞれの学習フェーズをマルウェアに関する攻撃セッションに適用する．学習期間は 2011 年 1 月 18 日から 1 月 24 日までである．この期間における CCC DATAset 2011 での 2 台（honey001, honey002）のパケットキャプチャログ（攻撃通信データ）を Snort に適用させ，アラートリストを生成した．なお，学習で利用するアラートにはマルウェアダウンロードと関連するローカルアラートも含まれる．ローカルアラートとの関連付けを行うために利用したマルウェアダウンロードログ（攻撃元データ）も学習期間内のものを利用した．このことより，攻撃元データと関連付けを行ったローカルアラートはマルウェアダウンロードアラートとして捉えることができる．また，学習フェーズにおいて生成されるローカルアラートの攻撃セッションもマルウェアダウンロードの攻撃セッションとして捉えることができる．

7.3.3 攻撃予測

2011 年 1 月 18 日から 1 月 24 日まで学習させたデータをもとに，2011 年 1 月 25 日から 1 月 31 日までの延べ 7 日間において攻撃予測を行う．7 日間それぞれの攻撃通信データを Snort に適用して出力された IDS アラートリスト（ライブストリーム）に対し予測運用を行う．ただし，IDS は本来マルウェアダウンロードをアラートとして出力しないため，ライブストリームでは前節で追加したローカルルールは利用しない．このローカルルールはマルウェア予測のための学習に利用するルールであり，運用時には使えないものであることに注意されたい．今回，CCC DATAset2011 における honey001 と honey002 の 2 台のハニーポットを対象とする．2 台のハニーポットそれぞれの IDS アラートリストを時系列データとして扱い，アラートが発生する度にアラートそれぞれの予測確率を計算した．

7.4 実験結果

7.4.1 マルウェア予測確率の推移

本研究では，Nexat をベースとした改良手法においてアラートそれぞれに出力された予測値をもとに，マルウェアダウンロードの予測確率に着目して評価を行った．図 7.1，7.2 に 2011 年 1 月 25 日の honey002，図 7.3，7.4 に 1 月 26 日の honey001 における時系列での予測確率の推移を示す．図 7.1，7.3 が Nexat[13] での結果であり，図 7.2，7.4 が改良手法の結果である．本論文では，未知のマルウェアによる攻撃と，侵入パターン (B) (8.2 節参照) が特に多かった両日に着目した．各図において，横軸の時間（0-24 時）に対し，縦軸はマルウェアダウンロード予測確率を示している．また，グラフ上の 1 本の線は予測成功の平均確率を示している．また，Nexat と改良手法共に予測確率に二極化が見られた．

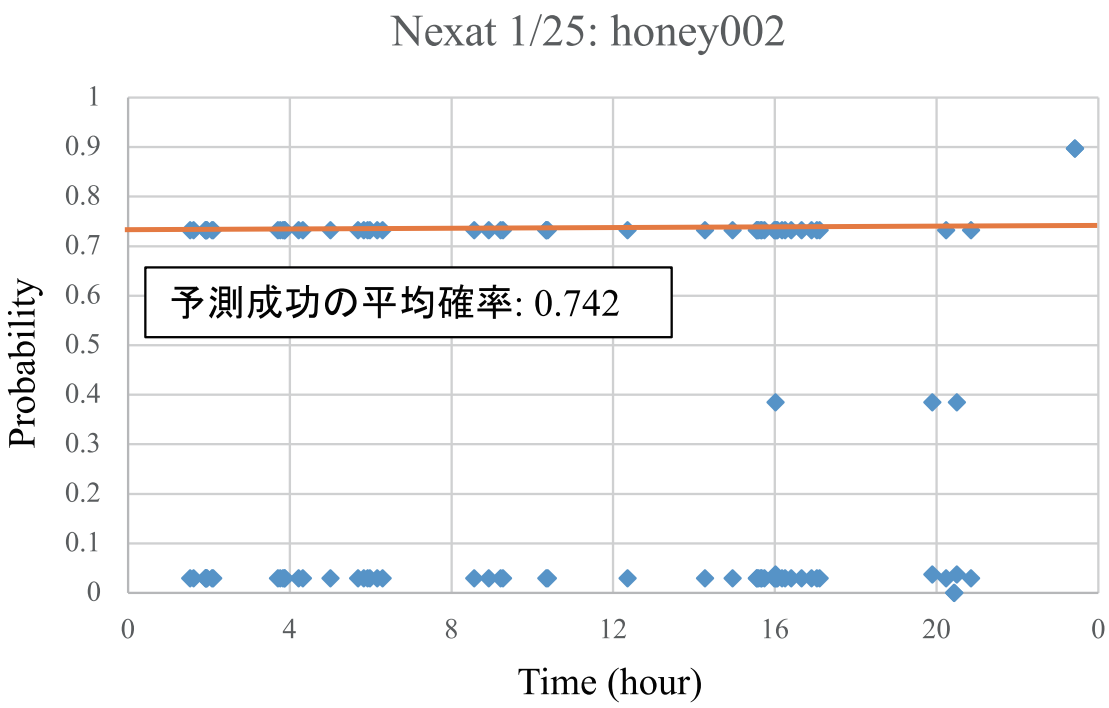

図 7.1: Nexat でのマルウェア予測: 2011/1/25 honey002

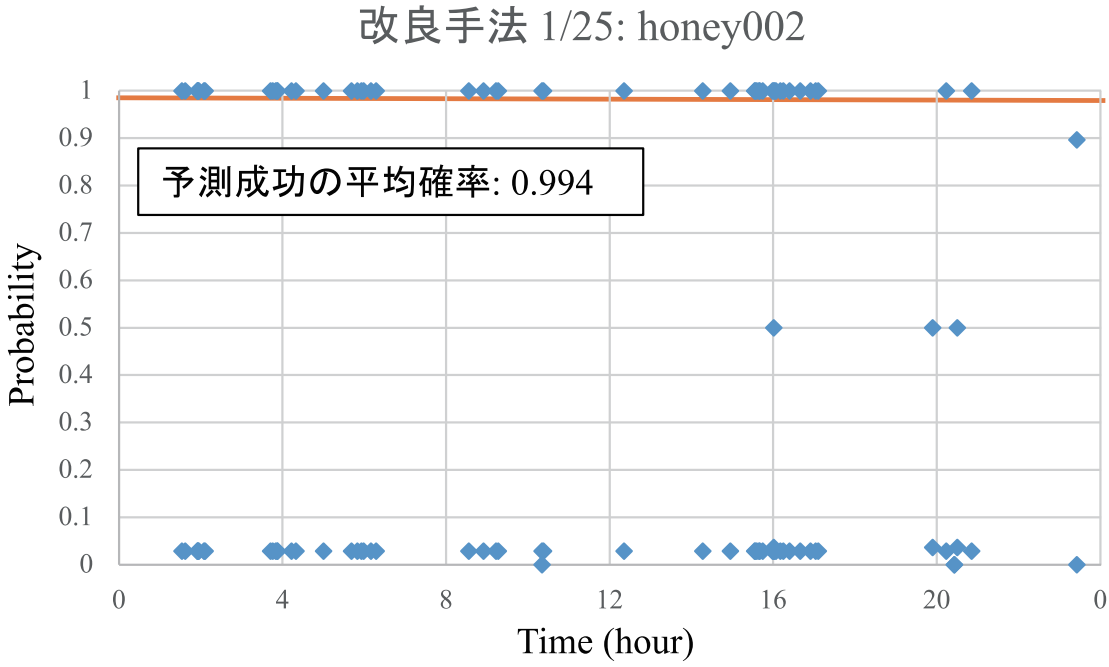

図 7.2: 改良手法でのマルウェア予測: 2011/1/25 honey002

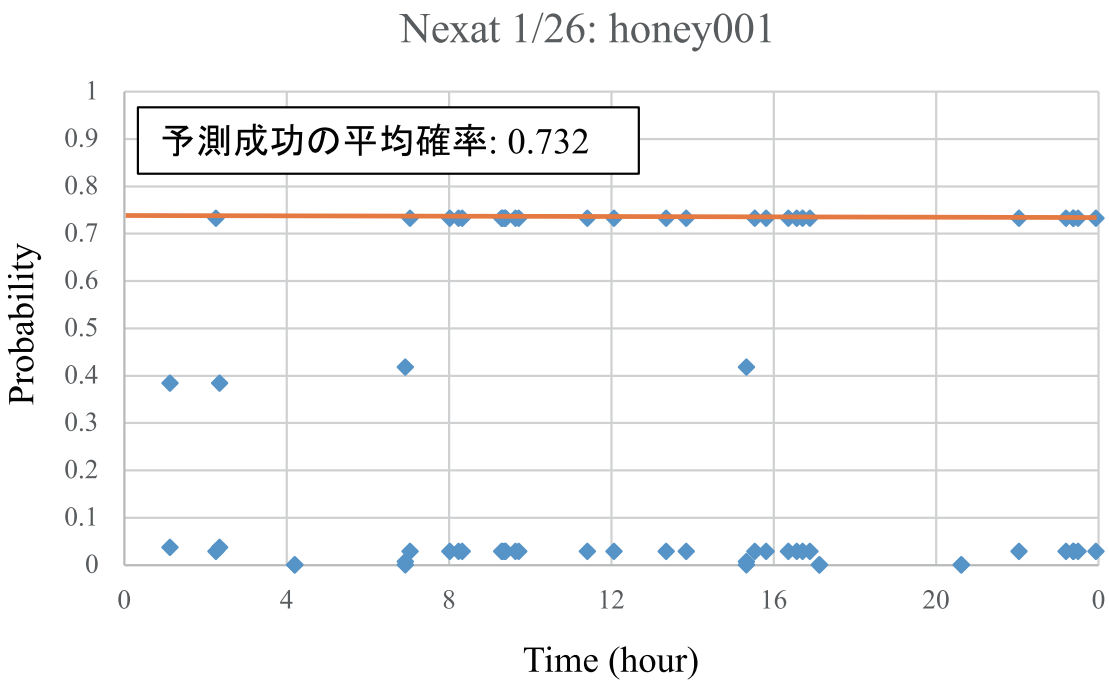

図 7.3: Nexat でのマルウェア予測: 2011/1/26 honey001

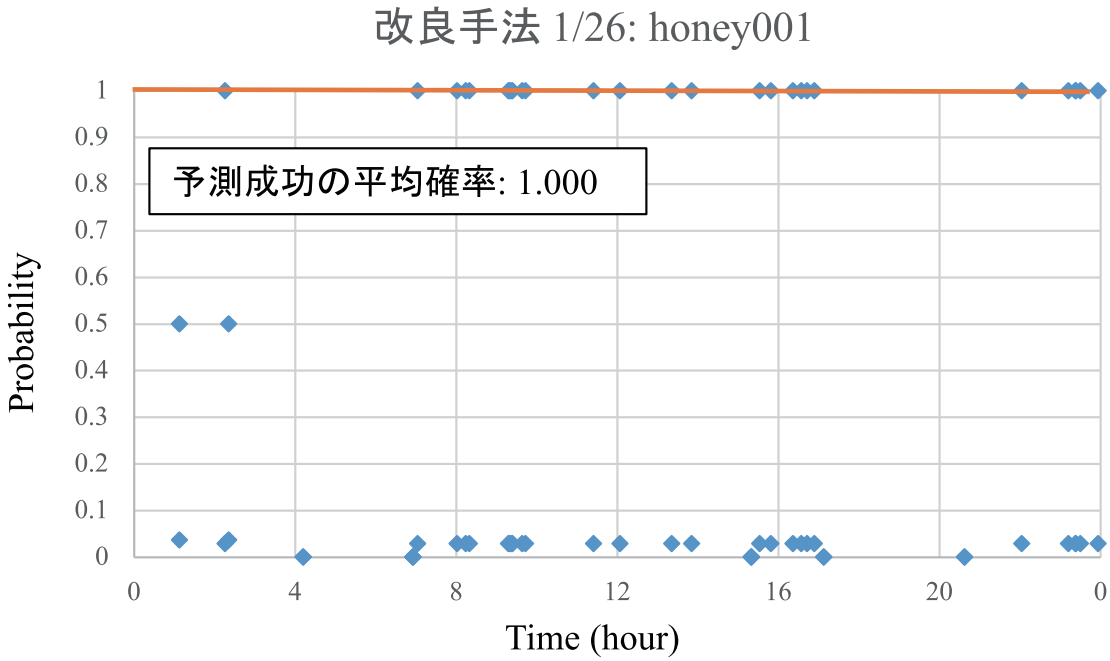

図 7.4: 改良手法でのマルウェア予測: 2011/1/26 honey001

このことは，予測確率の変化によりマルウェア侵入の判断が可能であると考えることもできる．

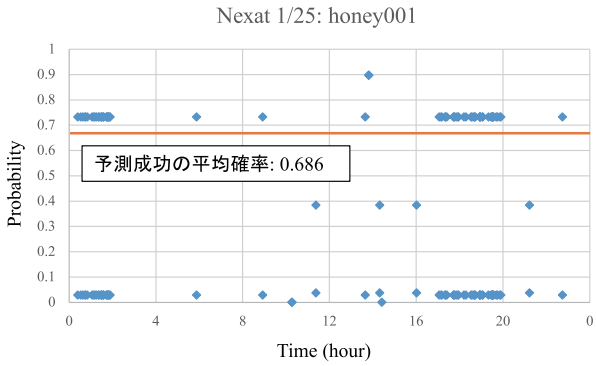

図 7.5: Nexat でのマルウェア予測: 2011/1/25 honey001

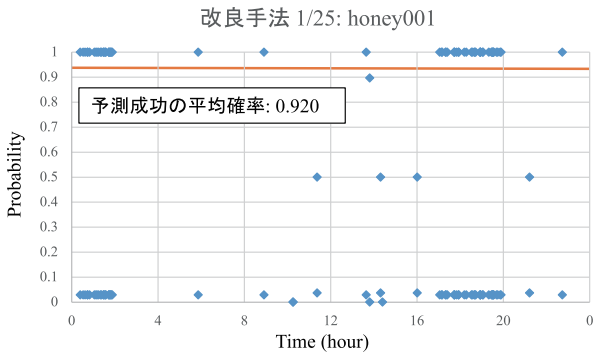

図 7.6: 改良手法でのマルウェア予測: 2011/1/25 honey001

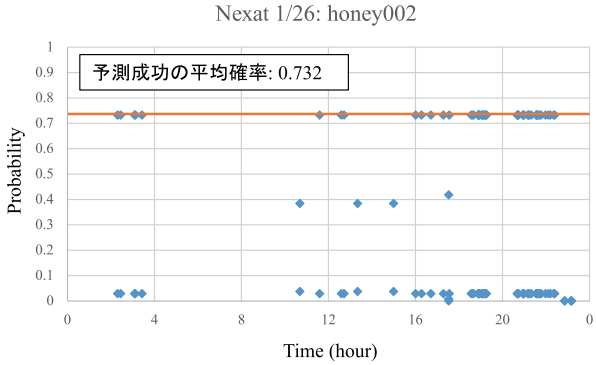

図 7.7: Nexat でのマルウェア予測: 2011/1/26 honey002

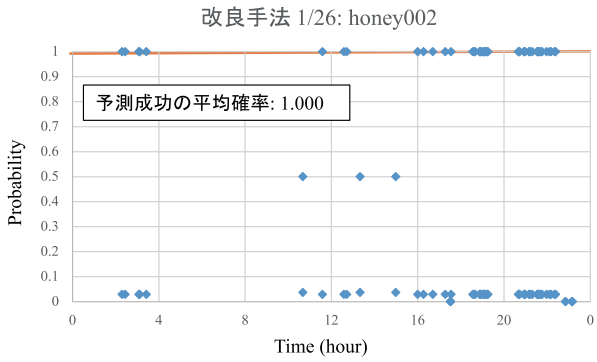

図 7.8: 改良手法でのマルウェア予測: 2011/1/26 honey002

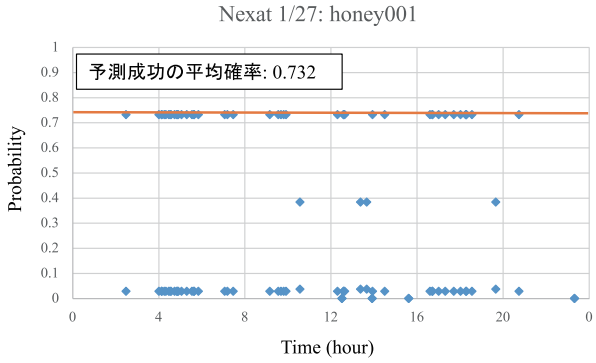

図 7.9: Nexat でのマルウェア予測: 2011/1/27 honey001

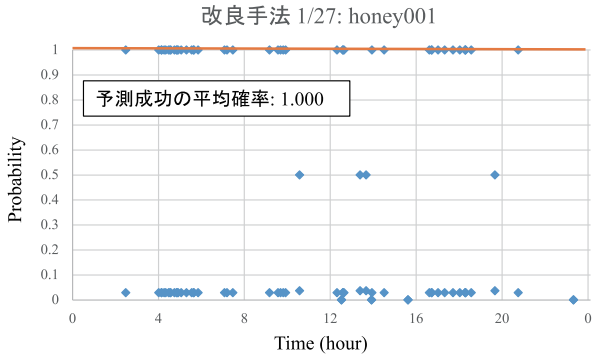

図 7.10: 改良手法でのマルウェア予測: 2011/1/27 honey001

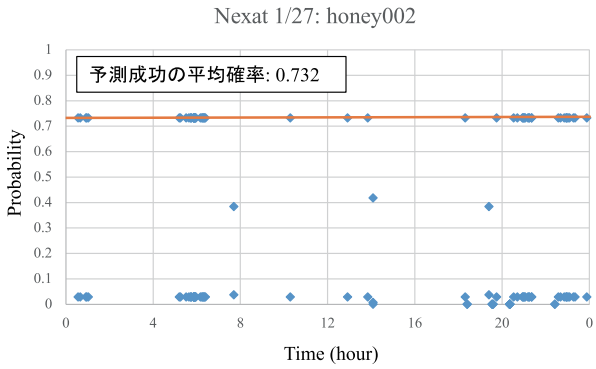

図 7.11: Nexat でのマルウェア予測: 2011/1/27 honey002

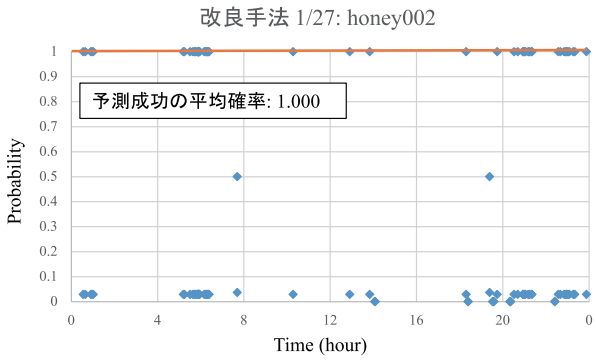

図 7.12: 改良手法でのマルウェア予測: 2011/1/27 honey002

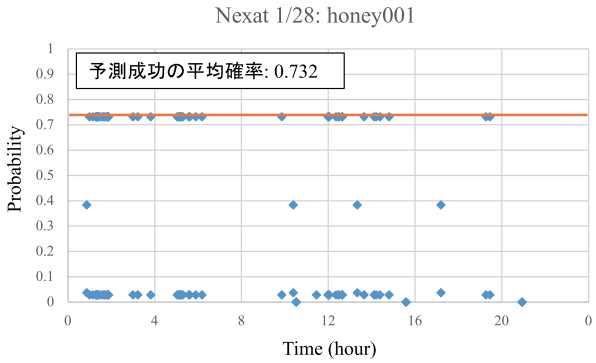

図 7.13: Nexat でのマルウェア予測: 2011/1/28 honey001

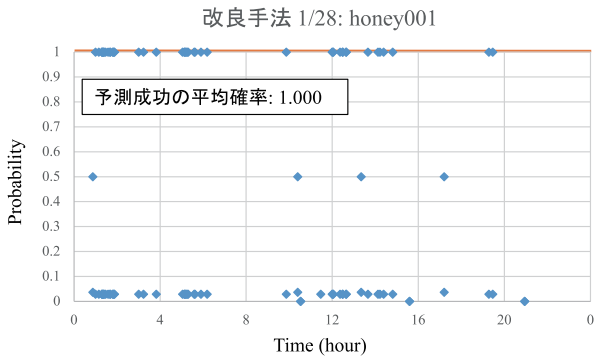

図 7.14: 改良手法でのマルウェア予測: 2011/1/28 honey001

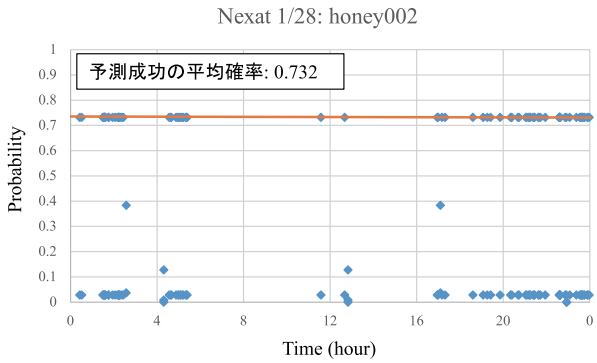

図 7.15: Nexat でのマルウェア予測: 2011/1/28 honey002

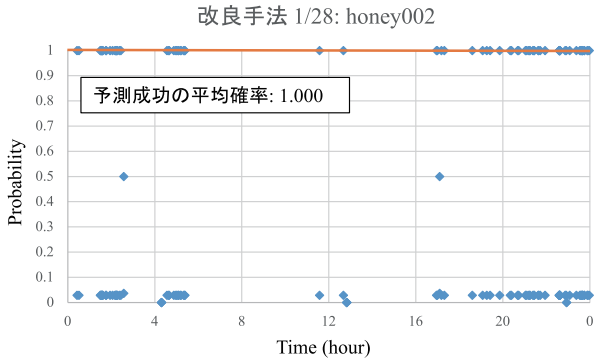

図 7.16: 改良手法でのマルウェア予測: 2011/1/28 honey002

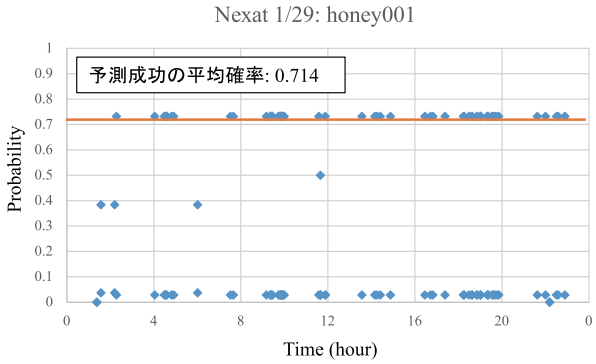

図 7.17: Nexat でのマルウェア予測: 2011/1/29 honey001

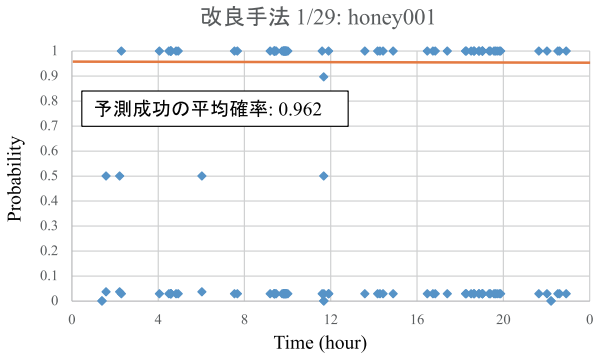

図 7.18: 改良手法でのマルウェア予測: 2011/1/29 honey001

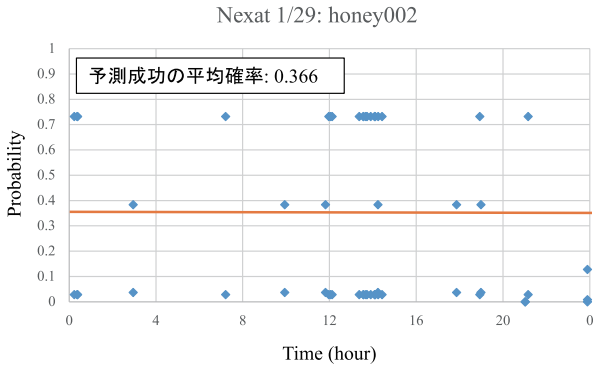

図 7.19: Nexat でのマルウェア予測: 2011/1/29 honey002

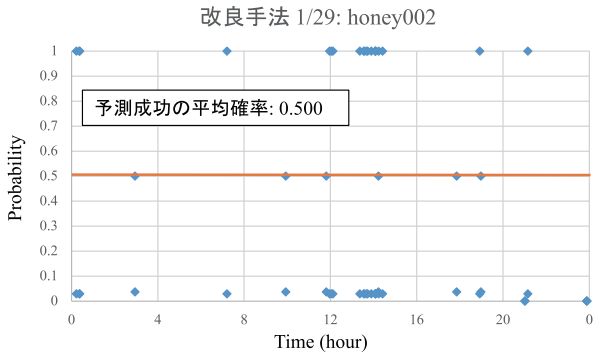

図 7.20: 改良手法でのマルウェア予測: 2011/1/29 honey002

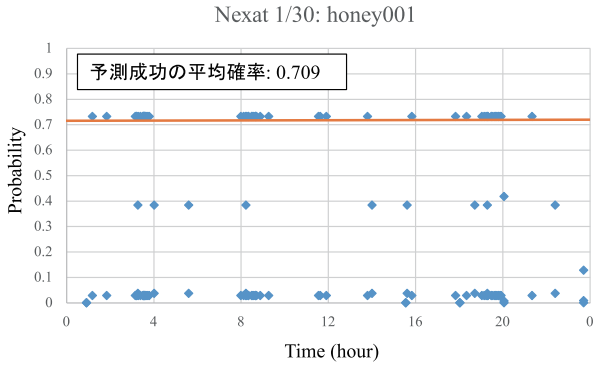

図 7.21: Nexat でのマルウェア予測: 2011/1/30 honey001

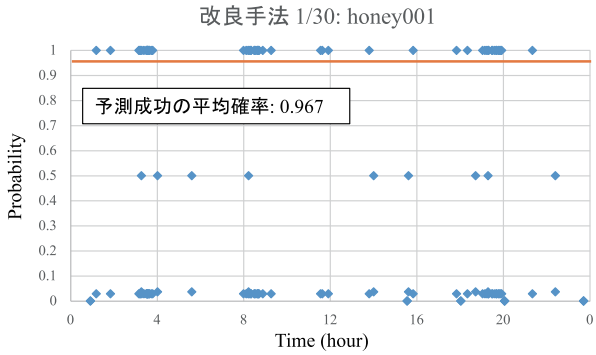

図 7.22: 改良手法でのマルウェア予測: 2011/1/30 honey001

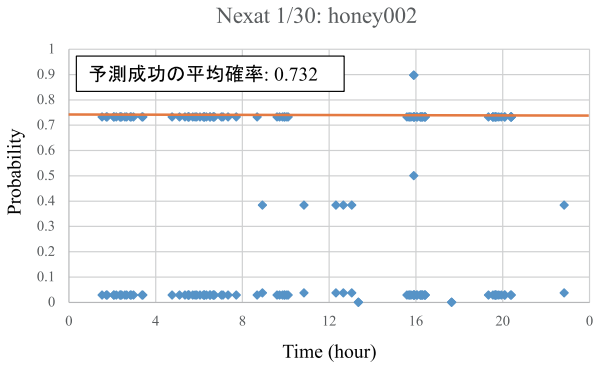

図 7.23: Nexat でのマルウェア予測: 2011/1/30 honey002

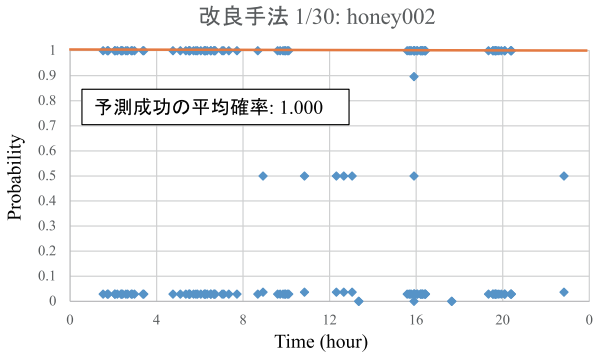

図 7.24: 改良手法でのマルウェア予測: 2011/1/30 honey002

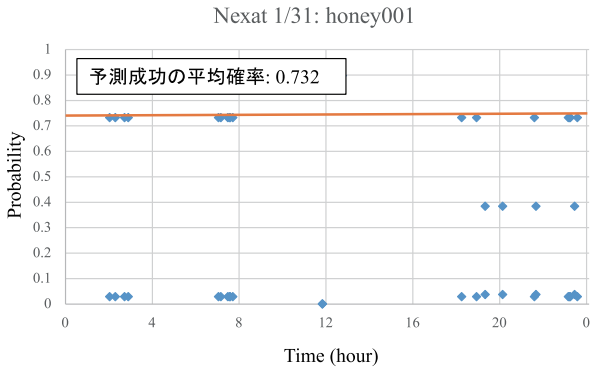

図 7.25: Nexat でのマルウェア予測: 2011/1/31 honey001

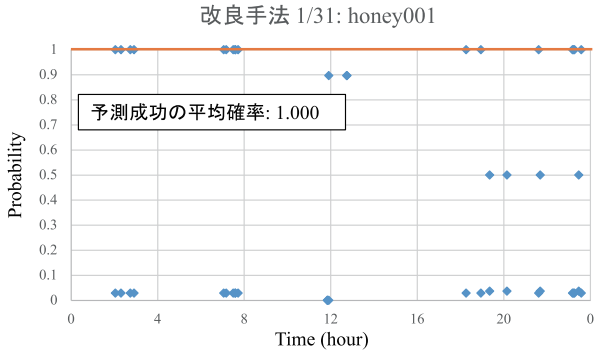

図 7.26: 改良手法でのマルウェア予測: 2011/1/31 honey001

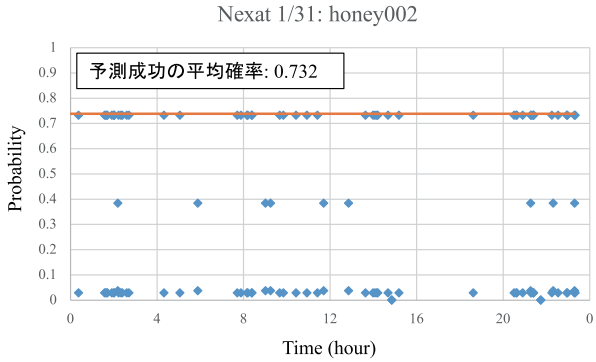

図 7.27: Nexat でのマルウェア予測: 2011/1/25 honey002

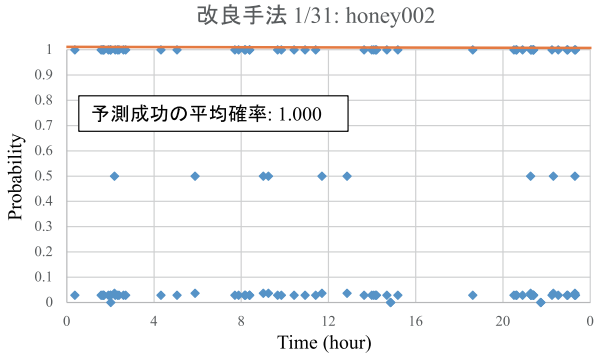

図 7.28: 改良手法でのマルウェア予測: 2011/1/25 honey002

表 7.1: 予測成功の平均確率（honey001）

	1/25	1/26	1/27	1/28	1/29	1/30	1/31
Nexat	0.686	0.732	0.732	0.732	0.714	0.709	0.732
改良手法	0.920	1.000	1.000	1.000	0.962	0.967	1.000

表 7.2: 予測成功の平均確率（honey002）

	1/25	1/26	1/27	1/28	1/29	1/30	1/31
Nexat	0.742	0.732	0.732	0.732	0.366	0.732	0.732
改良手法	0.994	1.000	1.000	1.000	0.500	1.000	1.000

7.4.2 予測成功の平均確率

ここで，予測成功の平均確率とは，ウィルススキャナで検知されたマルウェアに対して，その直前の攻撃セッションで導出された予測確率の平均値である．ただし，ウィルススキャナでマルウェアが検知されたものの，直前において攻撃セッションが生成されなかったものに関する予測確率は 0 とする．

1 月 25 日における honey002 での結果で比較すると（表 7.2），Nexat[13] では予測確率の平均値は 0.741 であるのに対し，改良手法では 0.994 となった．

Nexat では 3.2 節で説明したように，予測運用時にライブストリームから単位時間収集したアラート列をもとに PowerSet を生成し，生成されたアラート列を重みにアラートそれぞれの予測確率を導出している．いっぽう，改良手法ではライブストリームから単位時間収集したアラート列を直接攻撃セッションと考え，学習データ内にある攻撃セッションとの突き合わせを行うことにより，マルウェア予測確率を導出している．すなわち，現在発生している攻撃が，過去においてマルウェアと関連する攻撃であるか確認を行っている．このことは，予測日におけるマルウェアダウンロードの攻撃パターンが，過去において発生したマルウェアダウンロードの攻撃パターンとほぼ一致していることを示している．

第8章　考察

8.1　未知マルウェア

本研究のマルウェア予測実験で利用したデータセット [7] にはウィルススキャナにおいて未知（UNKNOWN）と定義されたマルウェアも含まれている．今回の予測実験で観測されたマルウェアのうち，既知のマルウェアと未知のマルウェアそれぞれの個数を表 8.1 に示す．

前述の通り，改良手法はマルウェアダウンロードが行われるまでの攻撃セッションを学習することにより，アラートベースでマルウェアの予測を行っている．すなわち，マルウェア本体が既知であるかを考慮することなく，過去にマルウェアダウンロードに利用された脆弱性情報のみでマルウェアダウンロードの予測を行っている．このことは，ウィルススキャナで “UNKNOWN” と出力された未知のマルウェアも予測を行えることを意味する．実際，1 月 25 日から 31 日までに検出された未知マルウェア 10 個は改良手法においてすべてダウンロードを予測することができた（表 8.2〜8.11）．

8.2　侵入パターン

マルウェアダウンロード攻撃を行う方法として以下の 3 つが考えられる（図 8.1）．

(A) IDS アラート攻撃元 IP アドレスとマルウェアダウンロード元 IP アドレスが同一．

(B) IDS アラート攻撃元 IP アドレスとマルウェアダウンロード元 IP アドレスが別．

(C) IDS アラートの発生なしでマルウェアダウンロードが行われた．

(A) の攻撃パターンはマルウェアダウンロードを行うための攻撃を行う攻撃者と，実際にマルウェアを与える攻撃者が同一の場合である．一方，(B) の攻撃パターンはマルウェアダウンロードを行うための攻撃者と，実際にマルウェアを与える攻撃者が別である場合である．また，Nexat と本研究の改良手法は IDS ベースである．IDS アラートを発生しない (C) の攻撃パターンでは予測確率の導出を行うことができない．7 日間の実験において観測された攻撃パターンを表 8.12 にまとめる．

また，未知マルウェアに着目すると，10 個の未知マルウェアは全て (B) による攻撃だった．このことは，未知マルウェアのダウンロードを行う前に，別の攻撃者がマルウェアダウンロードを行うための前処理を行っていることが明らかになった．

表 8.1: マルウェア検体数

		1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	既知マルウェア	20	11	12	7	13	15	7
	未知マルウェア	0	3	0	1	0	0	0
honey 002	既知マルウェア	16	12	12	19	2	18	16
	未知マルウェア	1	4	0	0	0	1	0

表 8.2: 予測成功・失敗マルウェア（Nexat: 閾値 0.2）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功マルウェア	既知	19	11	12	7	13	15	7
		未知	0	3	0	1	0	0	0
	予測失敗マルウェア	既知	1	0	0	0	0	0	0
		未知	0	0	0	0	0	0	0
honey 002	予測成功マルウェア	既知	16	12	12	19	2	18	16
		未知	1	4	0	0	0	1	0
	予測失敗マルウェア	既知	0	0	0	0	1	0	0
		未知	0	0	0	0	0	0	0

表 8.3: 予測成功・失敗マルウェア（改良手法: 閾値 0.2）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功マルウェア	既知	19	11	12	7	13	15	7
		未知	0	3	0	1	0	0	0
	予測失敗マルウェア	既知	1	0	0	0	0	0	0
		未知	1	0	0	0	0	0	0
honey 002	予測成功マルウェア	既知	16	12	12	19	2	18	16
		未知	1	4	0	0	0	1	0
	予測失敗マルウェア	既知	0	0	0	0	1	0	0
		未知	0	0	0	0	0	0	0

表 8.4: 予測成功・失敗マルウェア（Nexat: 閾値 0.3）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功 マルウェア	既知	19	11	12	7	13	15	7
		未知	0	3	0	1	0	0	0
	予測失敗 マルウェア	既知	1	0	0	0	0	0	0
		未知	0	0	0	0	0	0	0
honey 002	予測成功 マルウェア	既知	16	12	12	19	2	18	16
		未知	1	4	0	0	0	1	0
	予測失敗 マルウェア	既知	0	0	0	0	1	0	0
		未知	0	0	0	0	0	0	0

表 8.5: 予測成功・失敗マルウェア（改良手法: 閾値 0.3）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功 マルウェア	既知	19	11	12	7	13	15	7
		未知	0	3	0	1	0	0	0
	予測失敗 マルウェア	既知	1	0	0	0	0	0	0
		未知	1	0	0	0	0	0	0
honey 002	予測成功 マルウェア	既知	16	12	12	19	2	18	16
		未知	1	4	0	0	0	1	0
	予測失敗 マルウェア	既知	0	0	0	0	1	0	0
		未知	0	0	0	0	0	0	0

表 8.6: 予測成功・失敗マルウェア（Nexat: 閾値 0.4）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功 マルウェア	既知	18	11	12	7	13	14	7
		未知	0	3	0	1	0	0	0
	予測失敗 マルウェア	既知	2	0	0	0	0	1	0
		未知	0	0	0	0	0	0	0
honey 002	予測成功 マルウェア	既知	16	12	12	19	2	18	16
		未知	1	4	0	0	0	1	0
	予測失敗 マルウェア	既知	0	0	0	0	1	0	0
		未知	0	0	0	0	0	0	0

表 8.7: 予測成功・失敗マルウェア（改良手法: 閾値 0.4）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功マルウェア	既知	19	11	12	7	13	15	7
		未知	0	3	0	1	0	0	0
	予測失敗マルウェア	既知	1	0	0	0	0	0	0
		未知	1	0	0	0	0	0	0
honey 002	予測成功マルウェア	既知	16	12	12	19	2	18	16
		未知	1	4	0	0	0	1	0
	予測失敗マルウェア	既知	0	0	0	0	1	0	0
		未知	0	0	0	0	0	0	0

表 8.8: 予測成功・失敗マルウェア（Nexat: 閾値 0.5）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功マルウェア	既知	18	11	12	7	13	14	7
		未知	0	3	0	1	0	0	0
	予測失敗マルウェア	既知	2	0	0	0	0	1	0
		未知	0	0	0	0	0	0	0
honey 002	予測成功マルウェア	既知	16	12	12	19	2	18	16
		未知	1	4	0	0	0	1	0
	予測失敗マルウェア	既知	0	0	0	0	1	0	0
		未知	0	0	0	0	0	0	0

表 8.9: 予測成功・失敗マルウェア（改良手法: 閾値 0.5）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功マルウェア	既知	19	11	12	7	13	15	7
		未知	0	3	0	1	0	0	0
	予測失敗マルウェア	既知	1	0	0	0	0	0	0
		未知	1	0	0	0	0	0	0
honey 002	予測成功マルウェア	既知	16	12	12	19	2	18	16
		未知	1	4	0	0	0	1	0
	予測失敗マルウェア	既知	0	0	0	0	1	0	0
		未知	0	0	0	0	0	0	0

表 8.10: 予測成功・失敗マルウェア（Nexat: 閾値 0.6）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功マルウェア	既知	18	11	12	7	12	14	7
		未知	0	3	0	1	0	0	0
	予測失敗マルウェア	既知	2	0	0	1	0	1	0
		未知	0	0	0	0	0	0	0
honey 002	予測成功マルウェア	既知	16	12	12	19	2	17	16
		未知	1	4	0	0	0	1	0
	予測失敗マルウェア	既知	0	0	0	0	1	1	0
		未知	0	0	0	0	0	0	0

表 8.11: 予測成功・失敗マルウェア（改良手法: 閾値 0.6）

			1/25	1/26	1/27	1/28	1/29	1/30	1/31
honey 001	予測成功マルウェア	既知	19	11	12	7	12	14	7
		未知	0	3	0	1	0	0	0
	予測失敗マルウェア	既知	1	0	0	0	1	1	0
		未知	1	0	0	0	0	0	0
honey 002	予測成功マルウェア	既知	16	12	12	19	2	17	16
		未知	1	4	0	0	0	1	0
	予測失敗マルウェア	既知	0	0	0	0	1	1	0
		未知	0	0	0	0	0	0	0

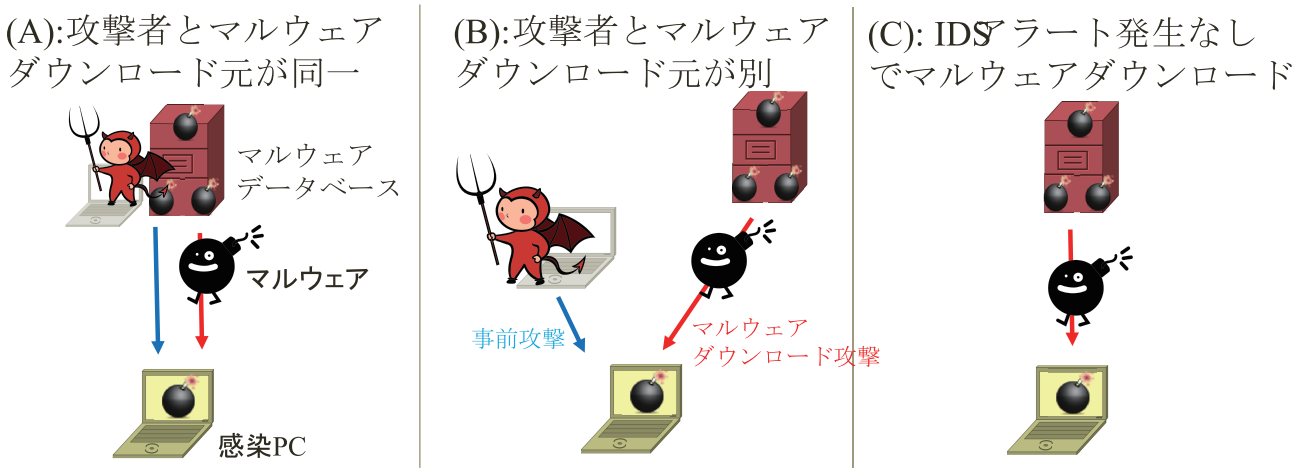

図 8.1: マルウェアダウンロード攻撃パターン

表 8.12: マルウェアダウンロード攻撃パターン

	1/25	1/26	1/27	1/28	1/29	1/30	1/31
(A)	23	16	18	24	11	28	12
(B)	13	15	7	3	2	9	12
(C)	1	0	0	0	1	0	0

8.3 誤検知

マルウェア予測において考えられる誤検知は　節の通りである．本実験では以下 2 つのマルウェアを予測することができなかった（4.2 節 (a) に相当）．

- 2011/1/25 14:24:30 honey001: BKDR_SDBOT.ZP
  （Rtsecar.exe）
- 2011/1/29 11:59:59 honey002: Mal_DLDER
  （fewh.exe）

上記 2 つのマルウェアは直前に前兆となるアラートが発生しなかったためマルウェア関連攻撃セッションを構築することができず，マルウェア予測確率を導出することができなかった（表 8.12 (C) に相当）．しかし，この 2 検体を除く全てのマルウェアダウンロードは予測確率の導出を行うことができた．

まず，深刻な誤検知である False Negative について比較する．予測成功の可否は導出された予測確率の大小によって決定する．具体的には，まず閾値を設定し，閾値より高い予測確率でマルウェアダウンロードが行われた場合，マルウェア予測成功（True Positive）とした．

表 8.15 に本実験における閾値 0.4 での誤検知率を示す．なお，IDS アラートの発生がなかった上記 2 つのマルウェアはどの閾値に設定しても False Negative に分類されることに注意されたい．

改良手法では過去に行われたマルウェアダウンロード前の攻撃行動とライブストリームとのアラートによる突き合わせにより予測確率を導出している．今回の実験において観測された攻撃のほとんどにおいて閾値以上の確率で予測を実現したことから，False Negative Rate が非常に低い値になることがわかった．

一方，それほど重要ではない False Positive についての比較も行う．表 8.14 に閾値 0.3 の誤検知率，表 8.17 に閾値 0.6 の誤検知率を示す．これら 2 つの閾値における False Negative Rate は Nexat と改良手法共に同一である．すなわち，False Negative と判断されたアラート数が同一であることを意味している．しかし，False Positive Rate を比較すると，閾値 0.3 では 1 月 26，30，31 日において，閾値 0.6 では 1 月 30，31 日において Nexat と比較して False Positive Rate が向上していることが見られた．これは，Nexat ではマルウェアダウンロードが発生しなかった時点での予測確率が閾値以上で導出されているものが，改良方

表 8.13: 誤検知率（閾値 0.2）

		1/25	1/26	1/27	1/28	1/29	1/30	1/31
False Positive Rate	Nexat	0.486	0.441	0.436	0.468	0.463	0.480	0.496
	改良手法	0.486	0.417	0.431	0.468	0.463	0.467	0.479
False Negative Rate	Nexat	0.027	0.000	0.000	0.000	0.063	0.000	0.000
	改良手法	0.027	0.000	0.000	0.000	0.063	0.000	0.000

表 8.14: 誤検知率（閾値 0.3）

		1/25	1/26	1/27	1/28	1/29	1/30	1/31
False Positive Rate	Nexat	0.486	0.441	0.436	0.468	0.463	0.480	0.496
	改良手法	0.486	0.417	0.431	0.468	0.463	0.467	0.479
False Negative Rate	Nexat	0.027	0.000	0.000	0.000	0.063	0.000	0.000
	改良手法	0.027	0.000	0.000	0.000	0.063	0.000	0.000

表 8.15: 誤検知率（閾値 0.4）

		1/25	1/26	1/27	1/28	1/29	1/30	1/31
False Positive Rate	Nexat	0.453	0.402	0.403	0.442	0.409	0.422	0.388
	改良手法	0.486	0.417	0.431	0.468	0.463	0.467	0.479
False Negative Rate	Nexat	0.054	0.000	0.000	0.000	0.063	0.027	0.000
	改良手法	0.027	0.000	0.000	0.000	0.063	0.000	0.000

表 8.16: 誤検知率（閾値 0.5）

		1/25	1/26	1/27	1/28	1/29	1/30	1/31
False Positive Rate	Nexat	0.453	0.378	0.398	0.442	0.409	0.418	0.388
	改良手法	0.486	0.417	0.431	0.468	0.463	0.467	0.479
False Negative Rate	Nexat	0.054	0.000	0.000	0.000	0.063	0.027	0.000
	改良手法	0.027	0.000	0.000	0.000	0.063	0.000	0.000

表 8.17: 誤検知率（閾値 0.6）

		1/25	1/26	1/27	1/28	1/29	1/30	1/31
False Positive Rate	Nexat	0.453	0.378	0.398	0.442	0.409	0.418	0.388
	改良手法	0.453	0.378	0.398	0.442	0.409	0.410	0.372
False Negative Rate	Nexat	0.054	0.000	0.000	0.000	0.125	0.054	0.000
	改良手法	0.054	0.000	0.000	0.000	0.125	0.054	0.000

式においては予測確率が閾値未満で導出されたことによる．このことから，改良方式では閾値によっては Nexat と比べ False Positive の誤検知が少なくなることがわかった．

第9章　おわりに

本研究では，IDS アラートをベースとした攻撃予測手法 Nexat について，マルウェアに対する有効性をマルウェア研究用データセットの一つである CCC DATAset 2011 を利用した予備実験を行った．予備実験を行う前に，本研究ではマルウェアに特化した学習を行うために以下の準備を行った．

- マルウェアダウンロード検知のための IDS ルールの作成．
- IDS の一つである Snort に攻撃通信データを適用させ，アラートログを生成
- マルウェアダウンロードアラート抽出

上記の準備によりマルウェアダウンロードを含むアラートログを生成し，Nexat の学習アルゴリズムに適用した．そして，マルウェアダウンロードを含まないアラートログに対し，マルウェアに特化した学習データを利用して Nexat を利用した予測実験を実施した．その結果，Nexat ではマルウェアに特化した学習データに対し，通常の IDS アラートのみでマルウェアの侵入を予測することが確認できた．さらに，ウィルススキャナにおいて未知マルウェアと定義されたものに対しても侵入の予測を実現できた．

また，マルウェアによる攻撃はロボットプログラムによるものが多いことに着目し，Nexat における予測アルゴリズムに対し，マルウェアに考慮した改良・簡単化を行い，性能評価を行った．具体的には，データ抽出並びに学習では Nexat のアルゴリズムを流用するが，攻撃予測において PowerSet を用いて予測確率を導出する部分を廃し，学習データを直接的に活用した予測確率の導出を行った．その結果，改良手法では Nexat と比べ予測精度の向上が見られた．未知マルウェアに対する予測も Nexat と同等の精度で行うことができた．

本研究の性能評価では予測運用においてもマルウェア研究用のハニーポットで収集された通信データを利用した．しかし，実際のネットワークではハニーポットのように外部から常時攻撃を受けていることは稀である．そのため，今後はハニーポットの通信データの他に，通常の通信データも含めた予測運用で性能評価を行いたいと考える．また，新しいマルウェア向けデータセットを用いた解析なども挙げられる．

謝辞

本研究を遂行するにあたり，指導教官である面和成准教授には常日頃より懇切丁寧にご指導していただいたばかりでなく，終始暖かな励ましやご助言を頂きました．また，本論文の不備をご指摘いただき，議論を展開する上で重要なヒントをいくつも頂きました．さらに，JAIST での生活の様々な面において多くのサポートをして頂いたこともありました．ここに心からの感謝の意を表します．

また, 様々な面でご支援を頂いた宮地充子教授にもこの場を借りて厚く御礼申し上げます．

さらに，Phuong Thao Tran 氏をはじめとした面・宮地研究室の諸氏には多くの面でお世話になりました．そして，私の両親をはじめとする，多くの方々からご支援を頂くことにより本研究を行うことができました．ここに記し，深く感謝いたします．

第10章　対外発表論文

[1] 森俊貴, 面和成, “Nexat を用いた攻撃予測に関する考察,” SCIS2013, 3C4-3, 2013.

[2] 森俊貴, 面和成, “ネットワーク通信アラートを利用した攻撃予測に関する評価・考察,” SCIS2014, 4C1-3, 2014.

参考文献

[1] “ボットネット概要,” JPCERT/CC 研究・調査レポート ボットネット研究資料, July 2006.

[2] C. Cipriano, A. Zand, A. Houmansadr, C. Kruegel and G. Vigna, “Nexat: A History-Based Approach to Predict Attacker Actions,” in *Proc. ACSAC 2011*, pp. 383-392, Dec. 2011.

[3] F. Cuppens and A. Miege, “Alert Correlation in a Cooperative Intrusion-Detection Framework,” In *Proceeding of the IEEE Symposium on Security and Privacy*, 2002.

[4] H. Deber and A. Wespi, “Aggregation and Correlation of Intrusion-Detection Alerts,” In *Proceeding of the International Symposium on Recent Advances in Intrusion Detection*, 2001.

[5] Christopher W. Geib and Robert P. Goldman, “Plan Recognition in Intrusion Detection Systems,” In *DARPA Information Survivability Conference Exposition (DISCEX)*, 2001.

[6] Daniel S. Fava, Stephen R. Byers and Shanchieh J. Yang, “Projecting Cyberattacks Through Variable-Length Markov Models,” *IEEE Trans. on Information Forensics and Security*, Vol. 3, No. 3, Sep. 2008.

[7] 畑中充弘, 中津留勇, 秋山満昭, “マルウェア対策のための研究用データセット ～ MWS 2011 Datasets ～”, CSS2011 (MWS2011), 2011.

[8] 川本研治, 市田達也, 市野将嗣, 畑田充弘, 小松尚久, “マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察,” CSS2011, p. 277-282, 2011.

[9] Zhi-tang Li, Jie Lei, Li Wang, and Dong Li, “A Data Mining Approach to Generating Network Attack Graph for Intrusion Prediction,” in *Proc. IEEE FSKD 2007*, 24-27 Aug. 2007.

[10] Peter Mell, Karen Kent and Joseph Nusbaum, “Guide to Malware Incident Prevention and Handling,” NIST Special Plublication 800-83, Nov. 2005.

[11] “Metasploit,” http://www.metasploit.com/.

[12] “Microsoft Security TechCenter,” http://technet.microsoft.com/en-us/security.

[13] 森俊貴, 面和成, “Nexat を用いた攻撃予測に関する考察,” SCIS2013, 3C4-3, 2013.

[14] 元田浩, 津本周作, 山口高平, 沼尾正行, データマイニングの基礎, オーム社, 2006.

[15] P. Ning, Y. Cui and D. Reeves, “Analyzing Intensive Intrusion Alerts via Correlation,” In *Proceedings of the International Symposium on the Recent Advances in Intrusion Detection*, 2002.

[16] P. Ning, Y. Cui and D. Reeves, “Constructing Attack Scenarios through Correlation of Intrusion Alerts,” In *Proceedings of the ACM Conference on Computer and Communications Security*, 2002.

[17] “nmap,” http://nmap.org/.

[18] “Snort,” http://www.snort.org/.

[19] 田中達也, 佐々木良一, “Snort ルールの組合せによるボット通信検知方式の確率と改ざんサイト自動検知システム DICE の改良,” CSS2011, p. 266-271, 2011.