A Port Scan Detection Method by Analyzing the Distribution Diagram of Accessed Ports

王, 璨
九州大学大学院システム情報科学府 | 九州先端科学技術研究所

馮, 尭鍇
九州大学大学院システム情報科学研究院 | 九州先端科学技術研究所

川本, 淳平
九州大学大学院システム情報科学研究院 | 九州先端科学技術研究所

堀, 良彰
佐賀大学全学教育機構 | 九州先端科学技術研究所

他

出版情報：Symposium on Cryptography and Information Security. 2014, 2014-01-21
バージョン：
権利関係：

SCIS 2014 The 31st Symposium on Cryptography and Information Security
Kagoshima, Japan, Jan. 21- 24, 2014
The Institute of Electronics, Information and Communication Engineers

ポートのアクセス数分布によるポートスキャン検知

A Port Scan Detection Method by Analyzing the Distribution Diagram of Accessed Ports

王 サン*, フォン ヤオカイ*, 川本 淳平*, 堀 良彰†, 櫻井 幸一*
Can WANG, Yaokai FENG, Junpei KAWAMOTO, Yoshiaki HORI, Kouichi SAKURAI

あらまし 近年，インターネットの普及と共に，人々の生活が益々便利になってきた．それに対して，ユーザが攻撃される危険性もある．多くの攻撃者は攻撃をする前に攻撃ターゲット及びその脆弱性を見つけるためにポートスキャンを実施する．ポートスキャンの早期検知ができれば，損害が抑えられることは言うまでもないが，関連の検知技術にはまだ課題が残ってある．例えば，閾値に基づいた手法がよく利用されている．しかし，閾値の決め方は簡単のことではない．そこで，本研究では，挙動に基づくポートスキャンの検知手法を提案する．本提案は学習データから通常時モードを抽出してそれを利用してポートスキャンを検知するので，事前に閾値を決める必要がなくなる．また，提案の核心部分となる学習アルゴリズムは既存のものと違ってパラメータを利用しないのでパラメータのチューニングは必要がなくなるところも本提案の大きいポイントになる．実験の結果により，本提案の学習アルゴリズムでの学習結果は目視の結果と殆ど同じであることが分かった．

キーワード ポートスキャン，異常検知，ネットワークセキュリティ

1 はじめに

近年，インターネットの利用率が益々高くなってきた．総務省によると，平成 24 年インターネットの利用率は 79.5%であり，近年の最大値となっている．それに対して，72.2%の利用者は個人情報の保護に不安を感じている[1]．そのため，安全・安心なインターネット環境を構築することが課題である．特に秘密情報が個人や企業が意識していないうちに不正者に窃取されると，大きな損失が発生してしまう．

不正者はターゲットの計算機へ侵入するために，ターゲットの脆弱性情報を収集することが多い．そこでは，ポートスキャンという技術が用いられる．ポートスキャンとは，ターゲットのポートをスキャンし，ターゲットで動作しているアプリケーションソフトや OS の種類を調べ，侵入口となりうる脆弱なポートがあるかどうか調べる行為である．ポートスキャンによってセキュリティホールを発見すると，本番の攻撃を行うことができる．

ポートスキャンは一般的に 4 つの種類がある．

・Vertical Scan：単一な始点 IP アドレスからある標的なホストに対して，そのホストの幾つか（もしくはすべて）のポートをスキャンする行為．

・Horizontal Scan：単一な始点 IP アドレスからある脆弱性があるポートに対して，複数のホストをスキャンする行為．

・Distributed Vertical Scan：複数の始点 IP アドレスから標的なホストの複数のポートをスキャンする行為．

・Distributed Horizontal Scan：複数な始点 IP アドレスからある脆弱性があるポートに対して，複数のホストをスキャンする行為．

そこで，攻撃の予兆であるポートスキャンを検知する研究が少なくない[2]．特に閾値という手法はよく利用されている．閾値の検知手法とはある一定時間の間に閾値を超える回数のアクセスがあればそれをポートスキャンと判断する手法である．Feng[3]は Distributed Horizontal Scan に向け，トラフィックデータから幾つかの情報を抽出してネットワークの挙動に基づく検知手法を提案した．挙動に基づく手法とは過去の IP アドレスの通信状態を

* 九州大学システム情報科学研究院, 福岡県福岡市西区元岡 744 番地, /九州先端科学技術研究所, 福岡市早良区百道浜 2 丁目 1 番地 22 号 福岡 SRP センタービル 7 階

† 佐賀大学全学教育機構, 佐賀県佐賀市本庄町 1 番地, /九州先端科学技術研究所, 福岡市早良区百道浜 2 丁目 1 番地 22 号 福岡 SRP センタービル 7 階

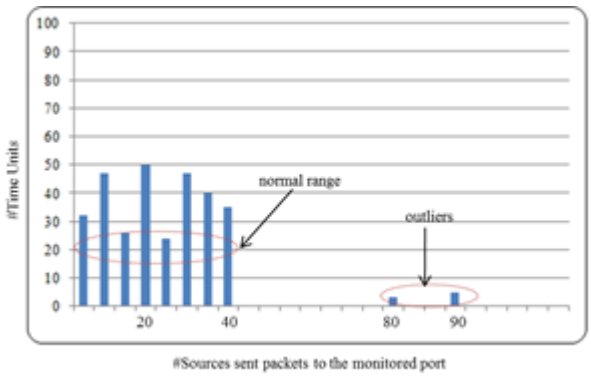

図 1. あるポートの度数分布

参考として学習し，新たなトラフィックを分析するという手法である．例えば，通常 1 つの時間単位内にあるポートへアクセしたユニークな始点 IP アドレス数が平均 20 である場合に，ある時間単位において，いきなり 400 始点 IP アドレスからアクセスがあれば, 攻撃をされていると判断する．

図 1 は，ある 1 つのポートに対して，横軸はそのポートへアクセスした始点 IP アドレスの数を表している. 縦軸はその数が発生した回数を表している．例えばある時間単位 10 分内にこのポートが 10 個の IP アドレスからアクセスされると，横軸が 10 のところに回数 1 を増やす. この図を見ると, このポートが殆どの時間で 40 種類以内の IP アドレスからアクセスされていることが分かる. そのため， 80 と 90 のところは異常であると考えられる. モデルを用いて説明すると，0 から 40 まではグループ 1 とし，80 から 90 まではグループ 2 とする．グループ 2 はグループ 1 まで距離が遠く, 比較的面積も小さいため, 異常値だと言っても過言ではない．したがって，通常の範囲は 0 から 40 までという結果になる. 即ち，この場合では閾値が 40 を決定し, 閾値以上の回数アクセスがあると異常と判断する．

しかし問題なのは，目視によって閾値を判断すると効率が悪くなる．できる限りアルゴリズムを作ってコンピュータが自動的に閾値を判断してくれる仕組みを考えたい．[3]は，距離 α とグループの面積 β 合わせて 2 つのパラメータを使う．簡単に説明すると，ますは図の右端から左へチェックする．もしあるグループの面積が総面積の β（例えば 10%）以下であり，且つ次のグループまでの距離は総距離の α（例えば 15%）以上の場合，このようなグループは特別とみなす．また，すべての特別なグループを削除すると，最後に残ったグループの右端を閾値とする．

しかし，この手法は 2 つの問題点がある．

(1) 学習結果がパラメータの変更により大きく変わる可能性がある．α と β を少々変わると，閾値はグループを飛んで大きく変化していく. 即ち，閾値はパラメータに対する敏感すぎる．

(2) ある 1 つのポートがアクセスされた数のみに注目している．例えば前述したように，学習結果が 40 の場合. あるポートが 40 以上の IP アドレスにアクセスされると，それは異常だと判断しアラートを発する．しかし，もしすべてのポートが僅かな数（例えば 10）の IP アドレスからアクセスされている場合，この時は閾値に超えないためアラートを発さない．実はこの挙動は無差別に全部のホストをスキャンして脆弱性なホストを探す行為であり，攻撃の一種だと考える．

以上の問題にも対応できるように，本研究は着目した対象を唯一のポートからすべてのポートに変えた. また, パラメータの影響を極力避けるために，新たなアルゴリズムを提案した．評価の結果により，このアルゴリズムの有効性を示した．

本稿の構成は以下のようになっている．2 節はポートスキャンに関する関連研究について説明する．3 節は本提案手法の詳細について説明を行う．4 節では実際のデータを使って，本手法を適用した結果を示し，また結果に対して評価と分析を行う．5 節ではまとめと今後の課題について述べる．

2 ポートスキャン

2.1 ポートスキャンとその種類

一般的には，コンピュータがネットワークと繋がると様々な UDP サービスと TCP サービスを使う．UDP サービスを提供しているポートは UDP ポートと呼び，TCP サービスを提供しているポートは TCP ポートと呼ぶ. 1 つのコンピュータは合わせて 65536 個のポートがある. それらのポートは一般的に 3 つの種類に分類されている. (i)Well-known ポート(0-1023)，(ii) 登録済みのポート (1024-49151)，(iii)ダイナミック/プライベートポート (49152-65535). 一般的には，ポートスキャンは直接コンピュータに損害を与えない．しかし，攻撃者はポートスキャンによってコンピュータの脆弱性を調べ, セキュリティホールを発見できる為, ポートスがスキャンされることはできるだけ避けたい．ポートスキャンは主に TCP ポートスキャンである. 即ち，TCP サービスを提供しているポートをスキャンすることである．UDP ポートスキャンも利用されている．しかし TCP ポートスキャンと比べて UDP ポートスキャンの信頼性が低い．その原因は UDP 通信方式と TCP 通信方式の違うからである．

ポートスキャンが以下のように分類できる.

(a) Stealth scan : 検出されないようにターゲットに TCP パケットを送る. 例えば SYN, FIN と NULL パケットなどが送信される.

(b) SOCKS port probe : SOCKS ポートは複数のコンピュータが同じインターネットコネクションをシェアすることサービスを提供する. ユーザーはこのポ

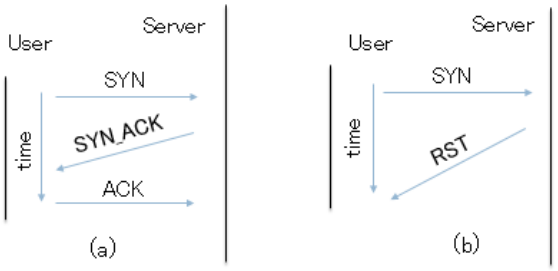

図 2. (a)宛先ポートは開いている, (b)宛先ポートは閉じている

スキャン. 現在最も利用されているポートスキャン技術の 1 つである.

(d) UDP scanning : UDP プロトコルによる UDP ポートをスキャンする. 前述したように, UDP ポートスキャンはあまり使わない.

2.2 ポートスキャンの検知手法

ポートスキャンの目的はポートの脆弱性をチェックして, コンピュータへ侵入することである. この目的を達成する為に, 攻撃者はたくさんのポートにアクセスしてみなければならない. 効率性も考えた上で, 短い時間内にできれば多くのポートをアクセスすることが望ましい. しかし現在の侵入検知システム (IDS) は益々高度になってきた. 短時間内に多くのアクセス挙動はすぐに検知される. また, アクセスの数だけではなく, 通信するときに送るパケットの種類にも監視している. 例えば良く利用されているハーフコネクションポートスキャン技術がある. 従来, 一般的なスキャン手法は 3 ウェィ・ハンドシェイクによる TCP スキャンである. 図 2 に示すように, 3 ウェィ・ハンドシェイクではユーザからサーバに TCP パケットを送り, サーバが TCP パケットを受信と TCP_ACK パケットをユーザに返信する. ユーザはこの TCP_ACK を見れば宛先のポートが開いていることが分かる. 最後では ACK パケットをサーバに送って, 3 ウェィ・ハンドシェイクは終わる. もし RST パケットが返信くれれば, それは宛先のポートが閉じていることを表す. しかしこのスキャン手法の問題点としてはログに記録が残ることである. ネットワークの管理者はログを見るとすぐ発見できる. このような状況を避ける為に, 現在はハーフコネクションポートスキャンを利用している. ハーフコネクションポートスキャンとは TCP パケットを送って, TCP_ACK パケットの返信がくれれば, ACK パケットを送らない事を指す. そうすれば正常な 3 ウェィ・ハンドシェイクは終わっていないため, 記録が残らない. また, TCP_ACK をすでに受信したので, 宛先のポート状況も知ることができる. この攻撃に対して, [4]はパケットの種類に基づく検知手法を提案した. もし TCP_ACK の後ろは ACK パケットではない場合, この時はハーフコネクションポートスキャン攻撃を受けた可能性があり, アラームを発する. また, 他の秘匿性があるポートスキャンに対しても, この手法は攻撃を検出できる.

前述したように, アクセス回数はある閾値に以上になると IDS に検出される可能性が高い. 攻撃者はこの状況を回避する為に, 意図的にアクセス頻度を下げて, スローポートスキャンという攻撃を行う. 例えば元々60 秒以内 30 回のアクセス頻度を, 600 秒かけて実行する. そうすると 60 秒平均 3 回のアクセスしかないため, 検出されにくいと考える. スローポートスキャンに対する幾つか検知手法が提案された. [5]はネットワークのトラヒック分類によるスローポートスキャン検知手法を提案した. この手法はネットワークに流れらパケットの種類と送信元 IP アドレス, 送信元ポート番号, 宛先 IP アドレス, 宛先ポート番号の組み合わせによって活動を分類した. 分類された活動は各事前に設定された活動パターンの特徴に当てはまるかどうかチェックし, どんなスキャンを判別できる.

本研究では, 複数の送信元 IP アドレスから, 無差別にポートをスキャンするような攻撃の検知を対象とする. 無差別ポートスキャンを用いてネットワーク上の任意のポートでもスキャンされる可能があるため, 提案手法はアクセスされた異なるポートに対して着目する.

3 提案手法

提案手法の流れを表 1 に表す. まず, 通常時モードを抽出するための学習データを収集する. そして, 必要な情報を抽出する. その後, 各時間単位にアクセスされたポート数を統計し, アクセスされた異なるポートの度数分布を作成する. 学習アルゴリズムを利用して, その度数分布から通常時挙動モードを抽出する. 通常時挙動モードを正しく抽出できれば最後の異常検知は簡単になる.

表 1. 提案手法の流れ.

Steps
1. 学習データの収集
2. データの抽出と統計
3.アクセスされた異なるポートの度数分布の作成
4. 通常時挙動モードの学習
5. 異常検知

3.1 データの抽出および統計

収集したトラヒックデータからポートスキャン検知に必要な情報を抽出する. 具体的に, 各時間単位にアクセスされた異なる終点ポートの数を統計する. しかし, 同じ時間単位に同じ始点から同一終点ポートへ重複アクセスした場合, 重複的に集計しない.

3.2 アクセスされた異なるポートの度数分布の作成

通常時モードを抽出するために, 3.1 で集計した各時間単位にアクセスされたポート数を利用して, そのポート数の度数分布を作成する. 例を図 3 で示す (ビンの幅 : 10) . 例えば, 4 番目のビンは, 時間単位ごとにアクセスポートの数が 30~40 である時間単位の数が 50 であることを意味している.

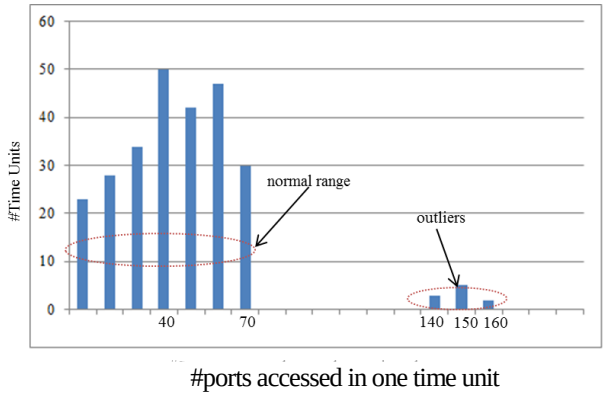

図 3. アクセスされた異なるポートの度数分布(例).

3.3 通常時挙動モードの学習

学習アルゴリズムの目的は，アクセスされたポートの度数分布から通常時挙動モードを抽出することである．例を図 3 で示す．図 3 で右側にある，遠く離れた 2 つのビンは，アクセスされたポート数が 130~160 である時間単位数が合わせて 10 個ぐらいしかないことを意味している．即ち，アクセスされたポート数は遥かに多くて稀にあることである．従って，それらのビンは異常と見なすべきである．異常のビンを削除して残りの部分は通常の範囲になる．本研究では，目視で通常時モードを決めるのではなく，学習で自動的にそれを抽出する．

既存研究[3]で度数分布から通常時モードを抽出するアルゴリズムを提案した．実験結果によりその学習アルゴリズムは有効である．しかし，そのアルゴリズムでは２つのパラメータが必要である．具体的に，度数分布の右側からビンを１個ずつ調べる．異常のビン（グループ）を判明する際，そのビン（グループ）の面積おおび次のビンとの距離に関する 2 つのパラメータを利用する．実は，その２つのパラメータのチューニングは難しい．本研究ではパラメータが要らない学習アルゴリズムを提案する．提案アルゴリズムは表２に表す．

表 2. 学習アルゴリズム.

Input: アクセスされた異なるポートの度数分布 Output: 通常時の挙動モード	
Initializing	Left_Pointer: 度数分布の最初の nonzero-bin-group の始まりを指す Right_Pointer: 度数分布の最後の nonzero-bin-group の終りを指す Span: 両端のビンの間の距離 Dist_right: Left_Pointer から次（右へ）の nonzero-bin-group の距離 Dist_left: Right_Pointer から次（左へ）の nonzero-bin-group との距離 Total_area: すべてのビン値の和 Area_right: Left_Pointer の次（右へ）の nonzero-bin-group の面積（ビン値の和） Area_left: Right_Pointer の次（左へ）の nonzero-bin-group の面積（ビン値の和）

Step 1
Right_Pointer を移動する
	While(Dist_left/Span > Area_left/Total_area){
	Move Right_Pointer to the start of the next left-hand zero-bin-group
	}

Step 2
Left_Pointer を移動する
	While (Area_right/Total_area>Dist_right/Span){
	Move Left_Pointer to the start of the next right-hand zero-bin-group
	}

Step 3
結果の判明
	If (Only zero-bin between Left_Pointer and Right_Pointer){
	Output Left_Pointer as the learning result
	} Else {
	Combine the two nonzero-bin-groups which on the left-side and right-side of Right_Pointer
	Go to Step 1.
	}

本提案では，2 つのポインター（Left_Pointer と Right_Pointer）を利用して，ポインターを移動することにより，最終の学習結果を決める．最初の時，度数分布の最左側と最右側のビンを指す．繰り返して，ある条件(Condition1, 後程説明する)が満たされれば, Condition1 が満たされなくなるまで，Left_Pointer は右に一歩ずつ移動する. 同じように, 繰り返して, ある条件(Condition2, 後程説明する)が満たされれば，Condition2 が満たされなくなるまで，Right_Pointer は左に一歩ずつ移動する．Left_Pointer に飛ばされたビンは異常ではないと判断できる．一方，Right_Pointer に飛ばされたビンは全部異常と見なすものである．

Case 1. 止まっている Right_Pointer と Left_Pointer の間にゼロビン（値がゼロである）のみの場合，即ち，この 2 つのポインターは同じ場所になる，または，その間にゼロのビンのみであれば，Light_Pointer の位置を学習結果とする．これは，2 つのポインターの間にゼロであるビンがあるかもしれないからである．

Case 2.そのほかの場合は，Right_Pointer の左にあるビングループ (後程説明する) を合併しなら, Condition2 を利用して左に移動する．Right_Pointer は移動できなくなると止まる．この時，1) 2 つのポインターの間にゼロビン以外のビンがなければ，上述のように，Left_Pointer の位置を学習結果と見なす． 2) 2 つのポインターの間にゼロビン以外のビンがあれば，Right_Pointer の位置を学習結果とする．これは，これ以上削除できるビンがないからである．

本研究で，ビングループとはビンの間に隙間なく（ゼロビンがない）最大のビンかたまりである．例えば，図 1 の場合は，ビングループは３つある．

上述した Left_Pointer と Right_Pointer の移動は，ビン単位でなく，ビングループ単位である．即ち，最初状態以外は，Left_Pointer は各ビングループの終わる所のみを選択の対象として左側から右に移動する一方，Right_Pointer は各ビングループの左端のところのみを選択対象として右側から左に移動する．

上述した Condition1 と Condition2 を説明する．Left_Pointer を移動する時，Condition1 を利用する．具体的に言えば，ビン全体のスパン（両端のビンの間の距離）に対する次のビングループとの距離の割合は，全体ビンの面積に対する次のビングループ面積の割合より小さければ Left_Pointer は右に一歩進む．次のビングループは異常として削除することができないからである．そうでない場合は，Left_Pointer が止まる．Right_Pointer は Condition2 を利用して右側から左に移動する．具体的に言えば，ビン全体のスパンに対する手前のビングループとの距離の割合は，全体ビンの面積に対する手前のビングループ面積の割合より大きければ Right_Pointer は左に一歩進む．手前のビングループは異常として削除することができるからである．そうでない場合は Right_Pointer が止まる．

上述したように，隣のビングループが合併される可能性がある．合併して新しくできたビングループにはゼロビンがあるようになる．Condition1 あるいは Condition2 を利用する際，ビングループ間の距離を計算する時にビングループ内のゼロビンも考える必要がある．本研究では，ビングループ内の距離と外の距離の和を利用している．

4　実験

4.1　実験データ

今回の実験は独立行政法人情報通信研究機構(NICT)よりいただいたダークネット観測データ（2011 年 6 月、7 月分）を利用する．また，時間単位とビンの幅が実験結果に対するどのような影響があるか確認したい為，時間単位は 10 分，30 分，60 分を設定し，ビンの幅は 250，500，750，1000，2000，3000 とした．実験の結果により，ビンの幅は結果に対して大きく影響しないことが分かった．そのため，次のセッションは代表としてビンの幅が 500，1000 の結果を示す．一方，時間単位の変更により，学習結果が特別な場合において変化するという結論を得た．

4.2　実験結果

図 4 から図 9 まで時簡単位とビン幅を変えることによって，学習結果に与える影響を探求する．図 10 は学習結果を用いてテストデータを検知してみる例である．

4.2.1　時間単位：10 分；ビン幅：500，1000

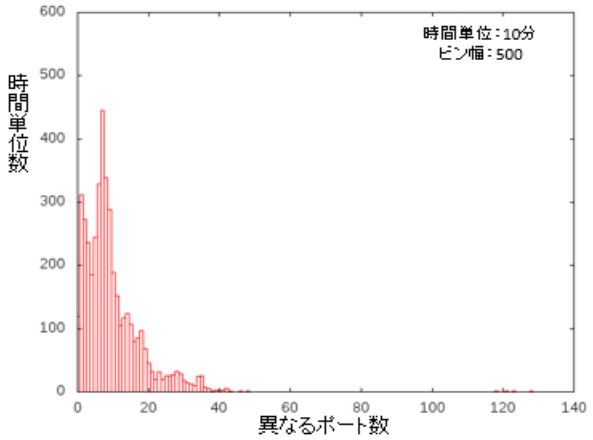

図 4．時間単位＝10 分，ビン幅＝500

学習アルゴリズムを利用して抽出した閾値：44*500 = 22000.

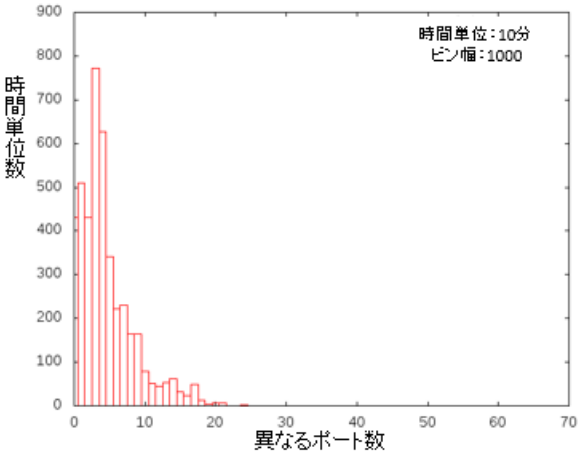

図 5．時間単位＝10 分，ビン幅＝1000

学習アルゴリズムを利用して抽出した閾値は 22*1000 = 22000.

4.2.2　時間単位：30 分；ビン幅：500，1000

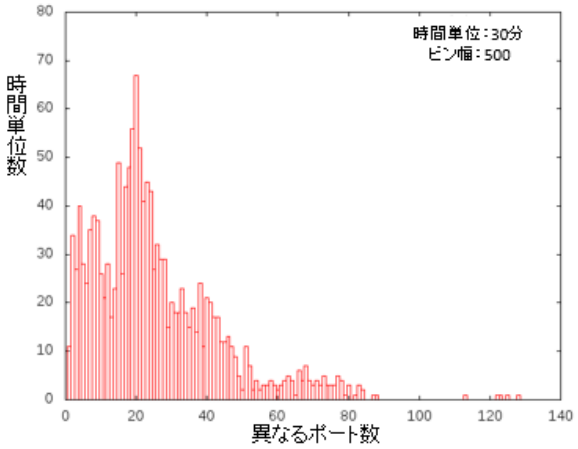

図 6．時間単位＝30 分，ビン幅＝500

学習アルゴリズムを利用して抽出した閾値は 81*500 = 40500.

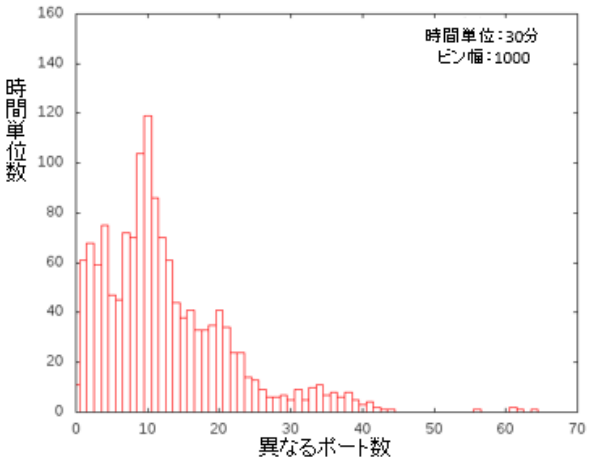

図 7. 時間単位＝30 分，ビン幅＝1000

学習アルゴリズムを利用して抽出した閾値は 45*1000 = 45000.

4.2.3 時間単位：60 分；ビン幅：500，1000

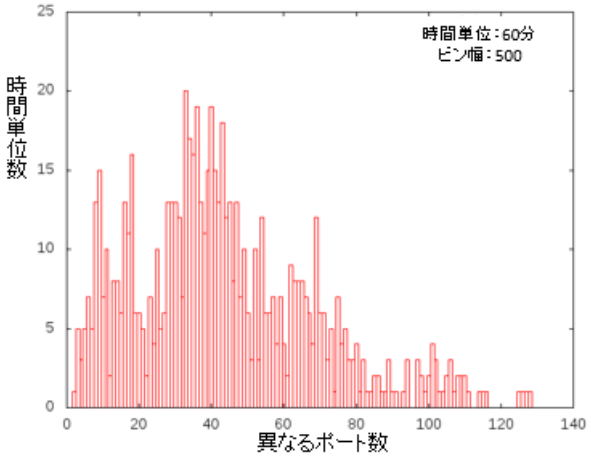

図 8. 時間単位＝60 分，ビン幅＝500

学習アルゴリズムを利用して抽出した閾値は 112*500 = 56000.

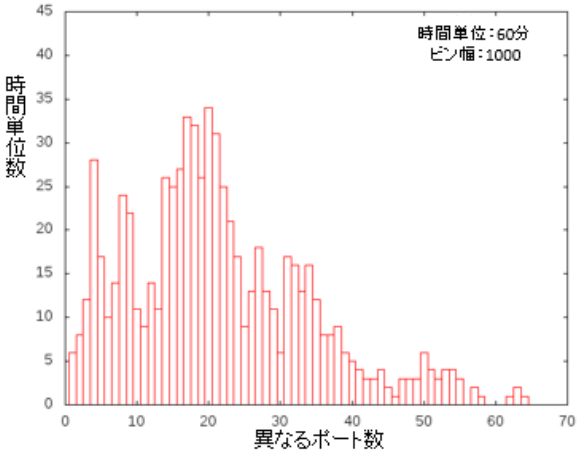

図 9. 時間単位＝60 分，ビン幅＝1000

学習アルゴリズムを利用して抽出した閾値は 56*1000 = 56000.

4.3 テストデータに適用して異常検知

今回は 2011 年 6 月のデータを学習し，2011 年 7 月のデータを検知してみる．結果は図 10 である．学習アルゴリズムを適用すると，得られた閾値は 45000 (45*1000) であり，7 月の異なるポート数は 45000 以上である場合すべて異常とみなす.

4.4 学習結果の分析

4.2 節の結果によると，ビンの幅が大きければ得られた閾値も大きくなる傾向がある．一方，時間単位を変更しても結果が変わらないことが分かる．ただし，時間単位を 30 分，ビンの幅を 500 または 1000 の場合（図 6 と図 7 を参照），学習結果が違った．原因を考えると，時間単位が 30 分でビンの幅が 500 の時（図 6 を参照），閾値(81)の隣は比較的に大きいグループがあるからである．目視で判断すると，閾値は 81 のところではなく，やや右の値(85)と判断する可能性がある．しかし実験のデータによると，横軸は 81 のとき縦軸の値 y[81]は 1 つの空ビンであり，その割合は 1/132=0.007575 になる．一方，y[82] から y[84] までの総和は 6，その割合は 6/1440=0.004167 である．数値を見ると，距離の割合は面積の割合の 1.8 倍以上である．即ち，この場合でも閾値を 81 とすることは適当である．目視で判断すれば間違えやすいところに対しても，この学習アルゴリズムは説得力がある結果を計算できる．また，閾値の付近に相対的に大きいグループがあれば，学習結果は必ずしも時間単位の変更によって変わらない.

5 まとめと今後の課題

5.1 まとめ

本研究では無差別ポートスキャン攻撃に対して，学習アルゴリズムを提案した．このアルゴリズムはパラメータがなくても閾値を決定できる利点がある．したがって，任意の学習データに対しても自動的に閾値の決定を実現できた．また，学習の結果は目視の結果とほぼ同じであることを示した.

5.2 今後の課題

3 節に学習アルゴリズムの詳細について説明した．そこで，ビングループが合併する時に，ビングループ内とビングループ外の距離の和を利用した．今後は合併後のグループ内の距離をもっと有効的に利用する．それに，様々なトラフィックデータを利用して本提案を詳しく検証する.

また，他のポートスキャン活動パターンも対応できるように，始点 IP アドレスと終点 IP アドレスも考慮する．更に，リアルタイムへの応用も考えていく.

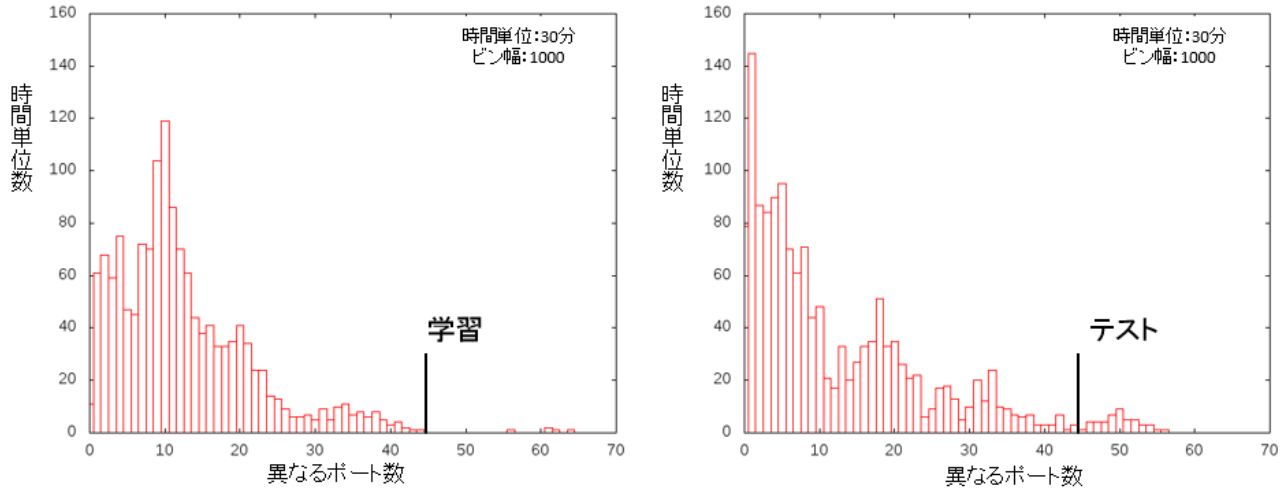

図 10.　6 月（左）を学習データとして 7 月（右）をテストする

謝辞

この研究の一部は，総務省による「国際連携によるサイバー攻撃の予知技術の研究開発」および科学研究費（基盤研究（C）No. 25330131)の支援を受けている．

本研究を実施するにあたり，独立行政法人情報通信研究機構(NICT)よりダークネット観測データの提供を受けた．ここに記して謝意を表します．

参考文献

[1] 総務省，「平成 25 年版情報通信白書」http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h25/

[2]M.H.Bhuyan, D.K.Bhattacharyya and J.K.kalita, “Surveying Port Scans and Their Detection Methodologies,” *The Computer Journal*, 2011, VOL 54, NO.10, pp. 1565-1581.

[3] Y. Feng, Y. Hori, K. Sakurai, J. Takeuchi, “A Behavior-Based Method for Detecting Distributed Scan Attacks in Darknets”, *Journal of Information Processing*, Vol 21, No. 3, pp. 527-538, 2013.

[4]M.Dabbagh, A.Ghandour, K.Fawaz, W.EL.Hajj and H.Hajj, “Slow Port Scanning Detection”, *International Conference on Information Assurance and Security (IAS)*, pp. 228-344, 7th, 2011.

[5] Joanne Treurniet,“A Network Activity Classification Schema and Its Application to Scan Detection,” IEEE/ACM TON, VOL 19, NO.5, 2011, pp. 1396-1404.