Vol.2017-CSEC-77 No.10
Vol.2017-IOT-37 No.10
2017/5/26

アクセススイッチにて取得するサンプリングパケットへの異なり数分析の適用

伊藤 昂平[1]　佐藤 聡[2,a)]　新城 靖[3]

概要：組織が利用するネットワークの内部にマルウェアが侵入した場合，情報流出などの多大な損害が発生する恐れがある．マルウェアが内部に侵入したことを早期に発見するためには，侵入したマルウェアは端末のスキャンといった内部の端末に対する通信を行うことから，アクセススイッチでそれを検出することが望ましい．しかし，アクセススイッチを通過するパケット数が膨大であるという問題がある．そこで本研究ではサンプリングによって減らしたパケットに対しても異なり数分析が有効であるかの検証を行うことを目的とする．実際にアクセススイッチにてサンプリングパケットと全ての通信パケットを取得して異なり数分析にかけた結果を比較すると，全パケットの結果と傾向が一致する場合と，より実際の通信の特徴に近くなる場合が見られ，サンプリングされたパケットを用いても通信の特徴が得られる可能性を示した．

キーワード：異なり数分析，不正・異常検出，パケットサンプリング

1. 序論

組織が利用するネットワークにおいて，内部の端末に外部から侵入されてしまうと，情報の流出などのリスクが存在する [1]．

マルウェアが内部に侵入されてしまった場合を想定し，可能な限りマルウェアによるリスクの発生を最小限にするため早期に対応することが不可欠であるといえる．そのためにはまず端末がマルウェアに感染したことを検出することが必要である．マルウェアに感染した端末は，感染対象となる他の端末を探すといった，ネットワーク内の他の端末に対する通信を行う [1]．つまり，こういったマルウェアの通信を見つけることが出来れば，それを行う端末がマルウェア感染していることを検出することが出来る．

マルウェアによる通信を発見するためにはルータ等のネットワークの境界，その下層に位置するスイッチ（上層側からコアスイッチ・ディストリビューションスイッチ・アクセススイッチ），端末に通信を監視する方法がある．しかし，ルータ上では内部端末同士の通信が通過しないために収集することが出来ない．また，端末上において通信を収集するには，その収集のための仕組みをあらかじめ端末に導入しておく必要があるが，ネットワークに接続する端末は外部から持ち込まれる場合もあり，常にすべての端末に導入していることは困難であるといえる．そこで本研究では末端のスイッチであるアクセススイッチを通過する内部ネットワークの端末の通信パケットの収集を行うことを考える．

ここで，アクセススイッチを通過する通信の量について考えると，ネットワークの規模が大きくなりアクセススイッチの台数が増加するほどパケット数は膨大になり，この大量の通信パケットをすべて収集することは非常に困難であると考えられる．そこで本研究ではサンプリングによって収集するパケット数を減らすことを考える．そこで，本研究ではインモン社が開発した sFlow[2] という技術を用いてパケットをサンプリングする．本研究ではアクセススイッチにてあるポートを通過する全てのパケットと，サンプリングパケットを取得し，それを対象として分析結果の比較により検証を行う．分析手法には本研究室で提案されてきた異なり数分析を用いる．

[1] 筑波大学 システム情報工学研究科 コンピュータサイエンス専攻
Graduate school of SIE, University of Tsukuba, Department of Computer Science
[2] 筑波大学 学術情報メディアセンター
Academic Computing and Communications Center, University of Tsukuba
[3] 筑波大学 システム情報工学域 情報工学域
Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba
[a)] akira@cc.tsukuba.ac.jp

2. 関連研究

2.1 内部ネットワークにおけるマルウェア検出の為の通信分析

Singh ら [3] はルータでの通信を分析することにより既存の，あるいは未知のワームの検出を試みている．ネットワークのトラフィックを観測し，多くのパケットに共通する同じ文字列を含む通信パケットを調査し，新たなワームとその通信パケットに含まれる文字列（シグネチャ）を自動的に識別する仕組みを作っている．Yen ら [4] はそれに加え，同じ外部ネットワークと通信している，類似のソフトウェアプラットフォームを利用するといったように，似た特徴を持つ通信の集合を定義することでボットネットやスパイウェアといった活動が少なく見つけにくいマルウェアの検出を試みている．

このように，境界部分では内部の端末と外部ネットワークとの通信を観測することが出来る．しかし，内部の端末同士が行う通信は境界を通過しないために観測することが出来ない．内部の端末同士が行う通信の情報を利用してマルウェアの検出を試みた研究に Gu ら [5] のものがある．Gu らは，端末間でやり取りされる TCP の SYN パケットと UDP パケットの情報を利用してワームの検出を目指した Destination-Source Correlation（DSC）アルゴリズムを提案している．DSC アルゴリズムでは，監視対象となるネットワーク内の端末が特定のポート i でパケットを取得し，ポート i 宛のパケットの送信を開始すると，そのホストを疑わしいとみなす．その後，疑わしいとみなしたホストが通常時から逸脱したレートでパケットを送信した場合，そのホストは感染したとみなす．

このような内部同士の通信が観測できるのは，通信パケットが通過する末端のスイッチであるアクセススイッチか，あるいは端末上のみである．しかし，端末上でパケットの収集を行う場合，ネットワーク全体を監視するにはすべての端末に収集のための仕組みが導入されている必要があり，ネットワークの規模が大きい場合や外部から端末の持ち込みが行われる場合に現実的ではないといえる．本研究では，アクセススイッチにおいて内部同士の通信を観測する．これにより，導入は端末よりも数の少ないアクセススイッチのみで良く，また外部から端末が持ち込まれた場合にもネットワーク等を変更することなく対応することが出来る．

端末と接するネットワーク装置上でマルウェアの検出を試みたものに，Jin ら [6] のものがある．Jin らは，携帯端末と接続するアクセスポイントにおいて，携帯端末に侵入したマルウェアの検出を試みている．Openflow による SDN を用いて，端末と接続したアクセスポイント（Openflow スイッチ）を通過するパケットを Openflow コントローラへ転送し，Openflow コントローラは 4 つのアルゴリズムを用いてアクセスポイント下のホストの通信の振る舞いが正常であるかどうかを判断する．正常であると判断される場合にはそのホストの通信を確立し，正常である基準に当てはまらない場合はすべてのパケットを破棄することでそのホストをネットワークから切断する．Jin らの研究では，Openflow スイッチを通過する全てのパケットを対象としており，多数のホストを想定した実験では 4 つのアルゴリズムを用いて検出を行う際に最大で 27.9%低下する結果となっており，より性能の良いサーバを用いれば処理時間の短縮を図れると述べている．本研究では，サンプリングにより処理するパケット数を削減することを考える．処理するパケット数を減らすことが出来れば，多数の端末がスイッチ下に接続する場合においても，必ずしも性能の高い計算機を用意することなく検出を行うことが出来ると考えられる．

2.2 異なり数分析

本研究では，アクセススイッチを流れるパケットを対象として，その通信を分析することを目的としている．本研究室で利用されてきた通信の分析手法の一つに，Shomura らが利用した異なり数分析が存在する [7]．パケットの送信元 IP アドレス（SIP），宛先 IP アドレス（DIP），送信元ポート番号（SPT），宛先ポート番号（DPT），プロトコル番号（PRT）を 1 つの入力アイテムセットとして扱い，5 つの値のうち n 個が一致しているアイテムセットを入力順に閾値個分集め部分集合を作る．その部分集合において，一致していない属性の種類数を異なり数という．異なり数の計測の様子を**図 1** に示す．図 1 の例は，閾値を 50，n=2 とし，SIP が IP1，PRT が TCP であるアイテムセットを閾値（50）個集めたところ，DIP の異なり数が 49，SPT の異なり数が 3，DPT の異なり数が 2 となった場合である．

図 1 のような出力が得られた場合，閾値が 50 であるのに対して DIP の異なり数が 49 と高く，DPT の異なり数が 2 と低いため，この SIP の端末がマルウェアに感染しており，他の端末に対するスキャンを行っている可能性があるという分析を行うことが出来る．

3. sFlow によるパケットサンプリング

sFlow[2] は，インモン社の技術である．元々はコアスイッチなどネットワーク全体のパケットが通過するような場所において，サンプリングにより効率的に，またリアルタイムにトラフィックの監視を行うために使用されている．

sFlow を用いたパケット収集の構成は，パケットをサンプリングする sFlow エージェントと，パケットを収集し分析するための sFlow コレクタからなる．sFlow を利用する際のネットワーク構成の例を**図 2** に示す．図 2 では sFlow コレクタは 1 台のみとなっているが，ネットワーク

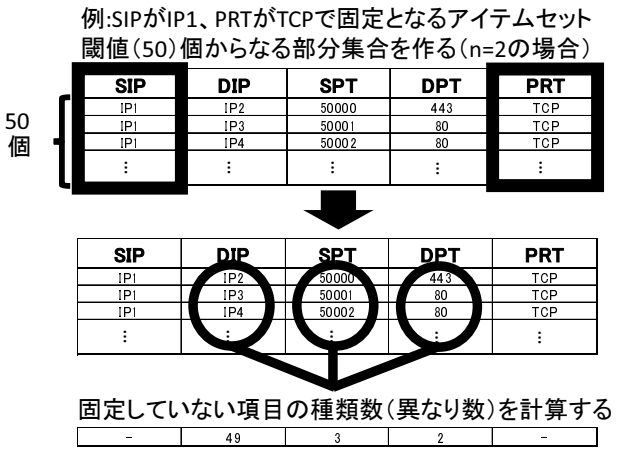

図 1　異なり数分析の流れ

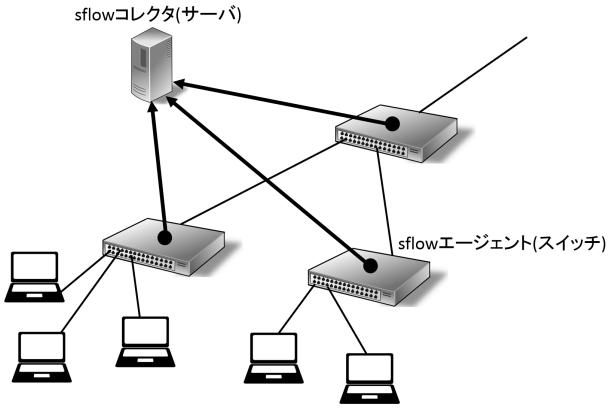

図 2　sflow を利用したネットワークの例

の規模や構成に応じて複数台に分けることも可能である．

sFlow の機能を持つスイッチは，スイッチの設定したポートを通過するパケットをサンプリングし，パケットのヘッダ情報をまとめたフローサンプルとその統計情報であるカウンタサンプルを作成し，それらを結合した sFlow パケットをネットワークを介して sFlow コレクタへ送信することが出来る．サンプリングはあらかじめおよそ何パケット毎にサンプリングするか（サンプリングレート）を設定し，長期的には全てのパケットとサンプリングされるパケットの数について $AllPackets/SampledPackets = SamplingRate$ を満たすように乱数により実際の間隔を定めて行う．sFlow コレクタは受信したサンプリングパケットの情報を利用して分析等を行う．本研究ではフローサンプルに含まれるヘッダ情報を用いて分析を行う．sFlow パケットのデータ構造は RFC3176[8] にバージョン 4 が，また [9] にバージョン 5 のものが記述されている．

4. パケットの収集・分析

4.1 sFlow パケットからの情報取得部分の実装

スイッチにおいてサンプリングされたパケットのヘッダ情報は，XDR を用いて変換され，UDP により sFlow コレクタに送信される [8]．sFlow コレクタでは，受信したパケットからサンプリングパケットのヘッダ情報を取り出し，**図 3** のデータ構造に格納して扱う．本研究では TCP と UDP のパケットに限定して分析を行う．

struct in_addr srcIp(4byte)	
struct in_addr dstIp(4byte)	
in_port_t srcPort(2byte)	in_port_t dstPort(2byte)
char protocol(1byte) / char check(1byte)	char padding(2byte)

図 3　分析用データ構造

SIP	DIP (異なり数)	SPT (異なり数)	DPT (異なり数)	PRT
IP1	49d	3d	2d	TCP

図 4　異なり数分析の出力

送信元 IP アドレスを srcIp，宛先 IP アドレスを dstIp，送信元ポート番号を srcPort，宛先ポート番号を dstPort，プロトコルを protocol にそれぞれ格納する．check は異なり数分析の際に用いるための値であり 4.2 項で述べる．padding はデータ構造全体を合計で 16 バイトに合わせデータ利用効率を良くするために含んでいる．

4.2 異なり数分析の実装

本研究では，Shomura ら [7] および Mori ら [10] のプログラムを改良し異なり数分析を実装した．用いるデータ構造を 5 つの 32 ビットのハッシュ値から図 3 のデータ構造に変更することで，主に以下の点の改良を行った．

- データ量を 20%削減した．
- 入力値を文字列からハッシュ値に変換する処理を省略した．

異なり数計測の結果となる出力の例をに示す．**図 4** は，図 1 と同様に，n=2 とし，SIP が IP1，PRT が TCP であるアイテムセットを閾値個集めたところ，DIP の異なり数が 49，SPT の異なり数が 3，DPT の異なり数が 2 となった場合である．異なり数である場合は数値の後に”d”をつけることで区別して出力する．

5. 評価

サンプリングしたパケットに異なり数が適用できることを検証するため，同じ時間帯の通信パケットすべてと，スイッチにおいてサンプリングした通信パケットをそれぞれ収集する環境を用意し，まずは収集の実験を行った．その実験の結果得られたパケット群を対象として，4.2 項で述べた異なり数分析用プログラムでそれぞれを分析することによって実験を行った．収集の実験に用いたネットワーク，実施方法とその評価，また機能に関する考察について述べる．

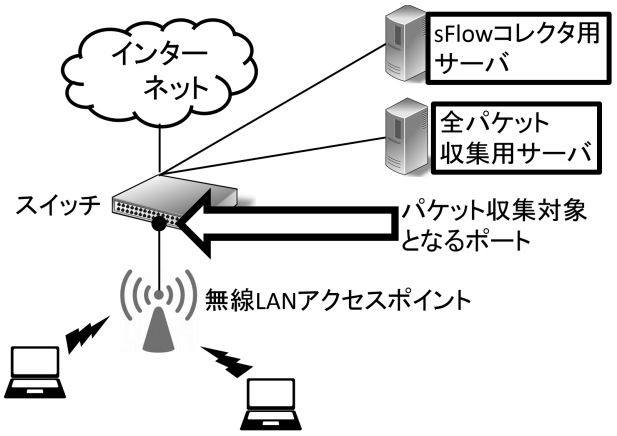

図 5　実験用ネットワーク構成概略図

表 1　使用した計算機の性能

	CPU	RAM	OS
sFlow コレクタ用サーバ	Xeon 3.47GHz	12GB	Ubuntu 14.04
全パケット収集用サーバ	Xeon 3.00GHz	64GB	CentOS 6.8

5.1　パケット収集環境の作成

実際にアクセススイッチにてパケットのサンプリングと全パケットの収集を行う為，アクセススイッチ下のアクセスポイントに端末が接続すると，その通信パケットを収集するようなネットワークを構築した．スイッチはジュニパーネットワークスの EX3300 イーサネットスイッチ [11] を用いた．ネットワーク構成の概略を**図 5** に示す．無線 LAN アクセスポイントに端末が接続すると，端末には 10.0.3.0/24 の範囲の IP アドレスが割り振られ，その通信パケットは図 5 に示したスイッチのポートを通過する．その際にパケットがサンプリングレート 100，つまりおよそ 100 パケットに 1 パケットの割合でサンプリングされ，sFlow パケットがインターネットを介して 4.1 項の sFlow コレクタ用プログラムが動作する sFlow コレクタ用サーバへと送信される．また，同じく収集対象となるポートを通過するパケットがスイッチのポートミラーリング機能によりコピーされ，全パケット収集用サーバへと送信される．この環境を用いて実験を行った．

sFlow コレクタ用，全パケット収集用サーバは**表 1** に示すものを用いた．また，収集したパケットの異なり数分析もそれぞれのサーバにおいて行った．

5.2　分析結果に関する評価

図 5 のネットワークを利用し，2017 年 1 月 5 日木曜日の 10 時から 18 時，2017 年 1 月 6 日金曜日の 13 時から 18 時の間パケットを収集した．2 日間の各サーバにおける収集パケット数は**表 2** のようになった．この集計においては，正確にサンプリングパケットが全パケットの 100 分の 1 とはなっていない．これは 3 項で述べた通り sFlow のサンプリング間隔が乱数で決定されることなどが原因と考えられるが，特定はできていない．

表 2　各サーバで収集されたパケット数

	全パケット	サンプリングパケット
1 日目	2396002	25740
2 日目	889682	10021
合計	3285684	35761

表 3　各 SIP 毎のパケット数

SIP	全パケット	サンプリングパケット
IP1	573	7
IP2	88020	894
IP3	98818	996
IP4	292783	2951
IP5	3778	30
IP6	322972	3261
IP7	117369	1166
IP8	238122	2352
IP9	20279	208
IP10	15648	158
IP11	435776	4364
IP12	53793	536
IP13	1964	18
合計	1689875	16941

また，各サーバにおける収集パケットについて，SIP が図 5 のアクセスポイントに接続した端末のものであるパケットに絞ると，各 SIP のパケット数は**表 3** のようになった．利用された IP アドレスを IP1，IP2，…，IP13 としている．

実験の結果を用いて，サンプリングしたパケットに対して異なり数分析が適用できているかどうか，つまりすべてのパケットを用いた異なり数分析から得られる端末の通信の特徴と同じような特徴をサンプリングしたパケットの分析結果からも得ることが出来るかどうか検証する．異なり数分析の閾値は 50 に設定した．表 3 に示される通り，通信量が少なく，十分なサンプリングパケットが得られなかった SIP が存在する．SIP が IP1，IP5，IP9，IP10，IP12，IP13 のものは十分な通信量に達していないとし，異なり数分析の比較評価は行わないものとする．

分析のために 4.3 節で述べたハッシュ表となるデータ構造の配列を 1024 の二乗個，1048576 個確保し行った．この時，Shomura らの手法と比較すると，配列 1 個あたり 4 バイトの削減が出来ていることから，合計で 4194304 バイト，約 4 メガバイトの削減が行えた．

収集したパケットを用いた異なり数分析の結果から，SIP が IPn（n=2，3，4，6，7，8，11）であり，PRT が固定値であるもの（TCP あるいは UDP），かつ SPT，DIP，DPT が異なり数であるものを抽出し，各 SIP について DIP と DPT の異なり数の組み合わせの度数を全体の度数の合計で割った割合（最大 1）を全パケットとサンプリングパケッ

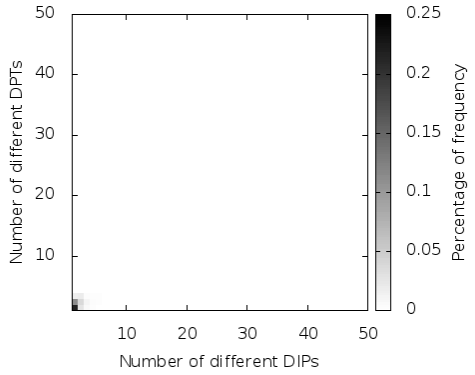

図 6 IP11 の全パケットの DIP と DPT の異なり数の組み合わせの度数を全体の度数の合計で割った割合の分布

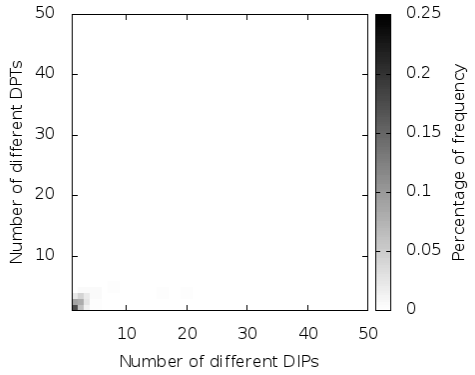

図 7 IP11 のサンプリングパケットの DIP と DPT の異なり数の組み合わせの度数を全体の度数の合計で割った割合の分布

トのそれぞれの場合において計測し，値の分布を比較したところ，3 種類の特徴が見られた．それぞれの特徴を示した SIP のうち一つについて，縦軸を DPT の異なり数，横軸を DIP の異なり数とし，度数の割合を黒色の濃度で表したグラフを全パケットとサンプリングパケットそれぞれのものを示し，各特徴について説明する．

1 つ目は，SIP が IP11 である通信に見られる特徴で，2 つのグラフの見た目が同じようになっている．IP11 について全パケットのグラフを**図 6**，サンプリングパケットのグラフを**図 7** に示す．図 6，図 7 のいずれに関しても，DIP，DPT の値が小さく，原点に近い位置に集中している．SIP が IP11 となるパケットのうち，DIP，DPT が同じパケットの数を計測し，パケットが多い順に上位 50 の DIP，DPT の組み合わせをとり，縦軸を通信パケットの数，横軸を通信パケットの多い順番の値としたグラフを**図 8** に示す．図 8 を見ると，IP11 からは限られた DIP，DPT のみにしかパケットが送信されていないことが分かる．これは特定の少数のプロトコルしか利用しておらず，かつアクセスしている先が少数であることを表している．よって SIP が IP11 の通信において，サンプリングパケットに異なり数分析を適用しても DIP，DPT の異なり数が共に小さくなる．そのため，全パケット，サンプリングパケットが共に同じ傾向を示したといえる．

2 つ目は，SIP が IP2，IP3，IP6，IP7，IP8 の通信に見られる特徴である．IP8 について，全パケットのグラフを**図 9**，サンプリングパケットのグラフを**図 10** に示す．図 9 と図 10 を比較すると，全パケットに比べ，サンプリングパケットの場合は DIP の異なり数のばらつきが大きくなっている．それぞれが異なる傾向を示したため，その理由について IP8 の全パケットを対象としてその通信の傾向を分析した．IP8 から送信されたパケットの DPT ごとの通信パケット数を計測し，縦軸をパケット数としたヒストグラムを**図 11** に示す．図 11 を見ると，IP8 は限られた DPT しか利用していないことが分かる．さらに，TCP において DIP，DPT が同じパケットの数を計測し，パケットが多い順に上位 50 の DIP，DPT の組み合わせをとり，縦軸を通信パケットの数，横軸を通信パケットの多い順番の値としたグラフを**図 12** に示す．図 12 を見ると，IP8 からは様々な DIP，DPT の組み合わせに対してパケットが送信されていることが分かる．すなわち少数のプロトコルしか利用していないが，そのプロトコルでアクセスしている先が多数あることを表している．実際に DPT の値を確認すると，最も多かったものが 80，次に多かったものが 443 であり，このユーザは web サーフィンをしていたと考えられる．また，それぞれの TCP セッションでは 1000 を超える数のパケットが含まれていることが確認できた．

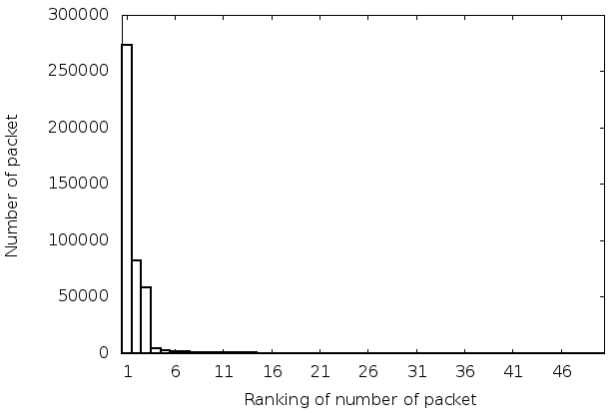

図 8 IP11 からの通信パケット数が上位 50 位となる DIP，DPT の組み合わせの通信パケット数

このことから，サンプリングパケットを用いた際に DIP のばらつきが大きくなった理由について次のように考察される．通常，同じ相手との TCP セッションは短期間に連続したパケットのやり取りとなっている．このとき，全パケットの場合では異なり数分析に用いる部分集合に，その短期間に連続した同じ相手との通信パケットが多く含まれ，DIP の異なり数が大きくなる．一方，サンプリングパケットの場合では，パケットがおよそ設定したサンプリン

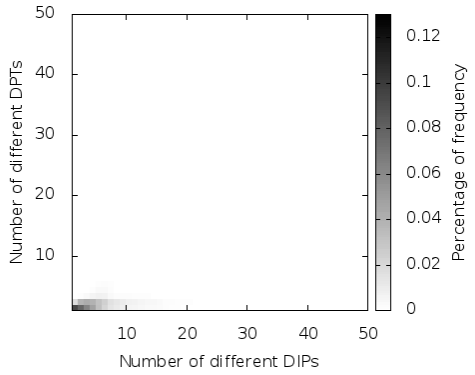

図 9 IP8 の全パケットの DIP と DPT の異なり数の組み合わせの度数を全体の度数の合計で割った割合の分布

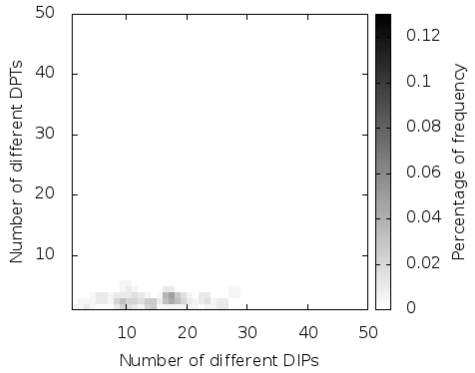

図 10 IP8 のサンプリングパケットの DIP と DPT の異なり数の組み合わせの度数を全体の度数の合計で割った割合の分布

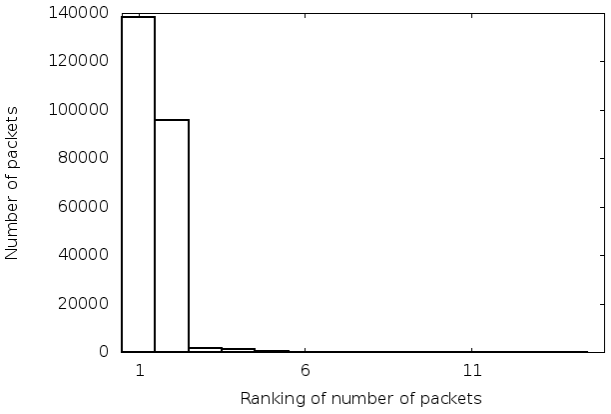

図 11 DPT ごとの IP8 との通信パケット数

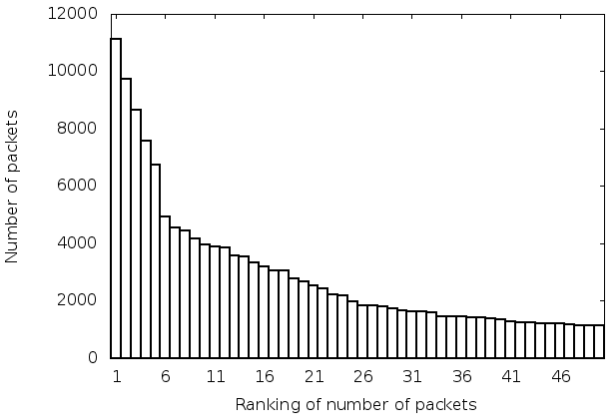

図 12 IP8 からの通信パケット数が上位 50 位となる DIP，DPT の組み合わせの通信パケット数

グレートの分だけ飛ばされて収集されることから，短時間にやり取りされる同じ相手との通信パケットは同じ部分集合に含まれにくく，DIP の異なり数は小さくなったと考えられる．それぞれのパケットを異なり数分析に適用した際に考えられる部分集合の様子を簡易的に**図 13** に示す．実際に IP8 の TCP パケットのやり取りを確認すると，同じ DIP，DPT の組み合わせとのやり取りが短時間に連続して行われていることが確認でき，これが正しいことが分かった．アプリケーションレベルの DDoS 攻撃はこのユーザの使い方に近くなるため，サンプリングパケットに異なり数分析を適用することによりそれらを発見しやすくなる可能性があることが分かった．

3 つ目の特徴は，SIP が IP4 の通信に見られた．IP4 の全パケットのグラフを**図 14**，サンプリングパケットのグラフを**図 15** に示す．全パケットでは DIP，DPT の異なり数がどちらもばらつきが大きく，サンプリングパケットではさらに大きくなっている．この IP4 の実際の通信の様子について調べるため，他の各 SIP と IP4 において，PRT ごとのパケット数を調査した．各 SIP の PRT ごとのパケット数を**表 4** に示す．表 4 を見ると，IP4 から送信された

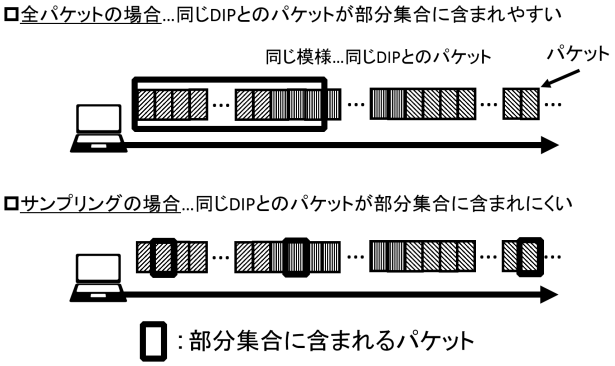

図 13 IP8 からの通信において部分集合に含まれるパケット

パケットの PRT について，TCP と比較して UDP が非常に多くなっていることが分かる．さらに，IP4 から送信されたパケットについて，DPT と PRT の組み合わせ毎にパケット数を計測したものを**表 5** に示す．これらの結果より，IP4 の端末は，主に UDP を用いて様々なサービスを利用して様々な相手と通信を行っていたと考えられ，DIP，DPT の異なり数は共にばらつきが大きくなるはずである．これに対し，全パケットの場合より DIP，DPT の異なり数のばらつきが大きくなったサンプリングの結果の方が，

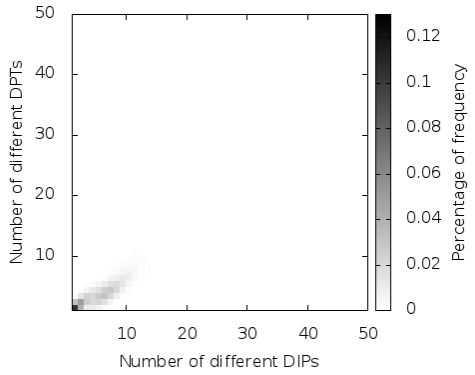

図 14 IP4 の全パケットの DIP と DPT の異なり数の組み合わせの度数を全体の度数の合計で割った割合の分布

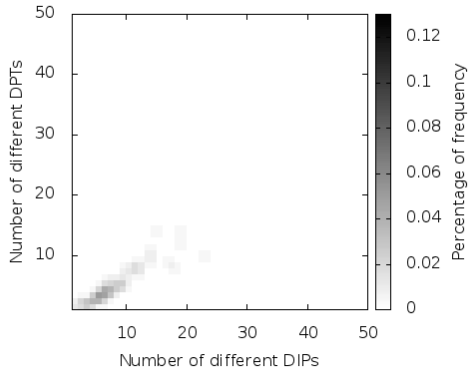

図 15 IP4 のサンプリングパケットの DIP と DPT の異なり数の組み合わせの度数を全体の度数の合計で割った割合の分布

表 4 各 SIP における PRT ごとのパケット数

SIP	TCP パケット数	UDP パケット数
IP2	42354	45666
IP3	97002	1816
IP4	29116	263632
IP6	304346	18416
IP7	113866	3157
IP8	222999	15054
IP11	286104	149642

実際の通信の特徴を表すことが出来ているといえる．

以上の結果より，サンプリングされたパケットに対して異なり数分析を適用すると，通信の特徴が下記の 3 種類に当てはまる場合には，DIP，DPT の種類数が少ない項目は全パケットと同じように異なり数が小さくなり，種類数が多い項目は異なり数のばらつきが大きくなるため，全パケットと比較してより特徴が分かりやすく出力されることが分かった．

- 少数のプロトコルを用いて少数の相手と通信している
- 少数のプロトコルを用いて多数の相手と通信している
- 多数のプロトコルを用いて多数の相手と通信している

よって，上記の 3 種類の通信について，サンプリングパケットに異なり数分析を適用しても通信の特徴について知ることが出来る可能性があることが示された．

表 5 SIP が IP4 の DPT，PRT の組み合わせ毎のパケット数

パケット数	DPT	PRT	パケット数	DPT	PRT
140229	443	UDP	203	54584	UDP
75176	19302	UDP	203	54538	UDP
19907	53731	UDP	202	63022	UDP
17398	443	TCP	186	5223	TCP
8939	3478	UDP	169	55053	TCP
6311	50784	UDP	166	80	TCP
4303	56742	UDP	135	53035	TCP
2006	143	TCP	124	62440	UDP
1734	993	TCP	110	63559	UDP
1539	61224	UDP	108	14416	UDP
1148	4000	TCP	96	192	UDP
1098	61224	TCP	88	1350	UDP
987	50784	TCP	84	4796	TCP
914	53508	TCP	62	59197	TCP
910	3000	TCP	62	38562	TCP
746	53	UDP	55	53035	UDP
600	4098	UDP	53	137	UDP
593	60519	UDP	50	5351	UDP
553	17500	UDP	46	1041	UDP
528	5353	UDP	44	55053	UDP
502	33477	TCP	40	3838	UDP
458	53508	UDP	36	59525	TCP
400	3001	TCP	22	59197	UDP
373	1900	UDP	22	38562	UDP
372	51336	UDP	16	123	UDP
369	14416	TCP	14	1900	TCP
352	4000	UDP	11	59525	UDP
322	53815	UDP	11	4796	UDP
266	1350	TCP	9	4098	TCP
222	63559	TCP	9	1041	TCP
218	5228	TCP	8	17355	UDP
205	49807	UDP	6	22	TCP
204	60131	UDP	6	17500	TCP
204	58284	UDP	2	138	UDP
203	55247	UDP	1	8255	UDP

5.3 機能に関する考察

本研究にて，アクセススイッチを通過するパケットのサンプリングを提案した理由は，第 1 章で述べた通り，スイッチに接続する端末数に比例して増加するパケットを収集することが困難であると考えられるためだった．そこで，サンプリングでパケットを収集することでリアルタイムな分析が行えるのかどうか，本節では考察する．

5.2 節のパケット収集において，1 日目は 8 時間の収集を行った．この 1 日目の 8 時間分の全パケット，サンプリングパケットそれぞれに異なり数分析を適用し，その実行時間を確認したところ**表 6** のようになった．計算機は表 1 に示した全パケット収集用サーバを用いて実行した．

表 6　全パケットとサンプリングパケットにおける実行時間

	パケット数	実行時間（s）
全パケット	2396002	32.195
サンプリングパケット	25740	0.426

表 6 において，サンプリングパケットにおける実行時間は，全パケットの約 76 分の 1 となっており，サンプリングレートの値に従った 100 分の 1 とは異なる結果となった．これは，それぞれの実行時間にはプログラムの起動やファイルのオープンといった，パケット数に依存しない処理の時間が含まれているためであると考えられる．さらにパケット数が増加するにつれて，パケット数に依存する処理の時間も増加し，この固定の処理時間が無視できるようになり，処理時間は 100 分の 1 に近づくと考えられる．よってこの結果を単純に比較することは出来ないが，サンプリングを行うことによって，実行時間を大きく削減することが出来た．つまり，接続する端末数等の条件に合わせ，サンプリングレートを設定することによって，接続する端末数が多くなった場合でも，リアルタイムな分析が行える可能性があることを示した．

今後の課題としては，サンプリングレートを変更した場合に，実行時間や結果にどのような影響があるか調査することが挙げられる．

6. 結論

本研究では，内部のネットワークに侵入したマルウェアを検出するための手法として，端末同士の通信を観測することのできるアクセススイッチにおいて，通信パケットをサンプリングし異なり数分析に適用しても同じような分析結果が得られるかどうかを検証することを目的とした．実際にアクセススイッチ下に接続した端末の全てのパケットと，サンプリングしたパケットを収集するためのネットワークを構築してパケットの収集を行い，それぞれに対して異なり数分析を実行した結果を比較した．3 種類のいずれかの特徴を持つ通信の DIP，DPT について，種類数が少ない項目は全パケットと同じように異なり数が小さくなり，種類数が多い項目は異なり数のばらつきが大きくなり，全パケットと比較してより特徴が分かりやすく出力されることがわかった．このことから，前述の特徴を持つ通信において，そのサンプリングパケットに対して異なり数分析を利用しても通信の特徴について知ることが出来る可能性があることが示された．

さらに，前述の特徴を持つ通信において DIP，DPT のうち種類数が多い項目は，サンプリングした場合により異なり数のばらつきが大きくなって出力されることから，従来の全パケットを利用した異なり数分析によっても検出出来ていた，少数の DIP に対し多数の DPT に通信を行うポートスキャンに加え，例えば，多くの DIP に対して http など特定の DPT から get による通信を行うようなスキャンについてもサンプリングした場合は特徴が分かりやすい出力が得られると考えられる．また，分析の機能について，接続する端末数などに応じてサンプリングレートを変更することでリアルタイムに分析を行うことが出来る可能性があると考えられる．

今後の課題としては，サンプリングパケットの異なり数分析の結果において，異なり数のばらつきが多い出力がされた場合，それが通常の通信によるものか異常な通信によるものかの識別が必要である．また，本研究ではマルウェア等の異常な通信を行う端末の検出が可能かどうかについては検証できていないため，マルウェアによる通信を含むパケットを対象とした調査も今後の課題である．

参考文献

[1] National Institute of Standards and Technology: マルウェアによるインシデントの防止と対応のためのガイド，IPA（オンライン），入手先 ⟨https://www.ipa.go.jp/files/000025349.pdf⟩（参照 2017-4-17）.

[2] sFlow.org: Making the Network Visible, InMon Corp (online), available from ⟨http://www.sflow.org/⟩ (accessed 2017-4-17).

[3] Singh, S., Estan, C., Varghese, G. and Savage, S.: Automated worm fingerprinting, *6th conference on Symposium on Opearting Systems Design & Implementation*, Vol. 6, pp. 45–60 (2004).

[4] Yen, T.-F. and Reiter, M. K.: Traffic Aggregation for Malware Detection, *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 207–227 (2008).

[5] Gu, G., Sharif, M., Qin, X., Dagon, D., Lee, W. and Riley, G.: Worm Detection, Early Warning and Response Based on Local Victim Information, *Computer Security Applications Conference, 2004. 20th Annual*, pp. 136–145 (2004).

[6] Jin, R. and Wang, B.: Malware Detection for Mobile Devices Using Software-Defined Networking, *2013 Second GENI Research and Educational Experiment Workshop*, pp. 81–88 (online), DOI: 10.1109/GREE.2013.24 (2013).

[7] SHOMURA, Y., WATANABE, Y. and YOSHIDA, K.: Analyzing the Number of Varieties in Frequently Found Flows, *IEICE Transactions on Communications*, Vol. 91, No. 6, pp. 1896–1905 (2008).

[8] InMon Corp: Network Working Group Request for Comments: 3176, , available from ⟨https://www.ietf.org/rfc/rfc3176.txt⟩ (accessed 2017-4-17).

[9] InMon Corp: sFlow Version5, , available from ⟨http://sflow.org/sflow_version_5.txt⟩ (accessed 2017-4-17).

[10] MORI, S., KATO, Y., SATO, A., and YOSHIDA, K.: Vicar in Network, 電子情報通信学会技術研究報告. IA, インターネットアーキテクチャ， Vol. 114, No. 286, pp. 21–26 (2014).

[11] ジュニパーネットワークス：EX3300 イーサネット スイッチ，，入手先 ⟨http://www.juniper.net/jp/jp/products-services/switching/ex-series/ex3300/⟩（参照 2017-4-17）.