4ZA-05

OpenFlow 環境における Packet-In メッセージの特徴に着目したポートスキャン検出に関する一検討

小野 大地 [†1]　和泉 諭 [†1] 阿部 亨 [†1,†2]　菅沼 拓夫 [†1,†2]

[†1] 東北大学大学院情報科学研究科　[†2] 東北大学サイバーサイエンスセンター

1 はじめに

ネットワークをソフトウェアで柔軟に制御・管理する仕組みである Software Defined Network (SDN) が注目され，その実装の OpenFlow が普及しつつある．OpenFlow は，トラフィックに関する統計情報の収集やスイッチ単位での通信の遮断が可能であることから，ネットワークのサイバーセキュリティ対策に有用であることが過去の研究で示されてきた [1] [2].

本研究では，サイバー攻撃の準備段階として行われるポートスキャンに着目する．OpenFlow 環境におけるポートスキャンの検出と防御に関する先行研究としては [1] があり，この手法を用いることで，スイッチの持つ統計情報からポートスキャンの検出と防御を容易に行える利点がある．しかしながら，この手法には，検出遅延とオーバーヘッド増加の課題がある．

この課題を解決するために，本研究ではスイッチが持つ統計情報だけでは無く，Packet-In メッセージの特徴を考慮することで，ポートスキャン検出と防御の効率化を行うことを目的とする．本稿では，Packet-In メッセージの流量からポートスキャンの発生を推定する手法の検討を行う．

2 関連研究

2.1 OpenFlow

OpenFlow は SDN を実現する技術の一つである．OpenFlow コントローラ（コントローラ）は OpenFlow スイッチ（スイッチ）に対して，どのパケットをどう処理するかを示したフローエントリと呼ばれる情報を書き込む．スイッチがパケットを受信した時，フローエントリに一致するものが存在しなかった場合，スイッチはコントローラに対して Packet-In メッセージを送ることで処理方法を問い合わせる．その後，新たなフローエントリがコントローラからスイッチに対して書き込まれる．また，OpenFlow スイッチはフローエントリ毎に統計情報を記録しており，受信パケット数や受信バイト数，フローエントリが作られてからの経過時間等の情報をコントローラで収集することが可能である．

2.2 ポートスキャン

通常，ネットワーク中のシステムに対する攻撃や侵入の前に，攻撃対象の脆弱性を調査するために攻撃者によるスキャンが発生する [3]．特に，攻撃対象のシステムが使用しているポートに関する情報を得るために行われるポートスキャンはメジャーな手法であり，ポートスキャンを行っている不審なホストを迅速に検出してネットワークから遮断することは，さらなる攻撃やマルウェアの感染拡大の防止に重要となる．

2.3 OpenFlow 環境におけるポートスキャン検出に関する研究

OpenFlow 環境におけるポートスキャン検出に関する研究としては文献 [1] がある．この手法では，各 OpenFlow スイッチの持つフローエントリの統計情報を数秒間隔で周期的に収集し，それらを分析することでポートスキャンが発生したかどうかを判定している．課題として，統計情報を一定間隔で収集しているため，ポートスキャンの発生から検出までに最大で収集間隔分の遅延が発生する点が挙げられる．また，全てのスイッチから統計情報を収集しているため，ネットワークを構成する OpenFlow スイッチ数の増加に応じて，トラフィック量やネットワーク I/O のオーバヘッドが増加する点が挙げられる．

3 提案手法

3.1 概要

本研究では，前章で述べた課題に対する新たなポートスキャンの検出手法の検討を行う．

一般にポートスキャンが行われると，短時間でこれまでの通信とは異なる複数のポートに対するパケットが発生する．OpenFlow 環境においては，スイッチが初めて受信する種類のパケットに対して Packet-In メッセージが発生することから，ポートスキャンが行われた場合，Packet-In メッセージの局所的な増加が発生すると考えられる．本研究では，この特徴に着目した手法を提案する．

提案手法のフローチャートを図 1 に示す．提案手法では，各スイッチからコントローラに対して送られる Packet-In メッセージをモニタリングし，流量の異常増加を起点に特定のスイッチから統計情報を収集し，ポートスキャンの検出処理を行う．この手法によって，全てのスイッチから周期的に統計情報を収集する必要が無くなるため，即座に検出が可能となり，トラフィック量やネットワーク I/O のオーバーヘッド増加も抑えられる．

3.2 Packet-In メッセージの流量のモニタリング手法

Packet-In メッセージには，Packet-In メッセージの発生源となったスイッチの識別子（DPID）と

A Study on Port Scan Detection Based on the Characteristics of Packet-In messages in OpenFlow Networks

Daichi ONO[†1], Satoru IZUMI[†2], Toru ABE[†1,†2], and Takuo SUGANUMA[†1,†2]

[†1]Graduate School of Information Sciences, Tohoku University

[†2]Cyberscience Center, Tohoku University

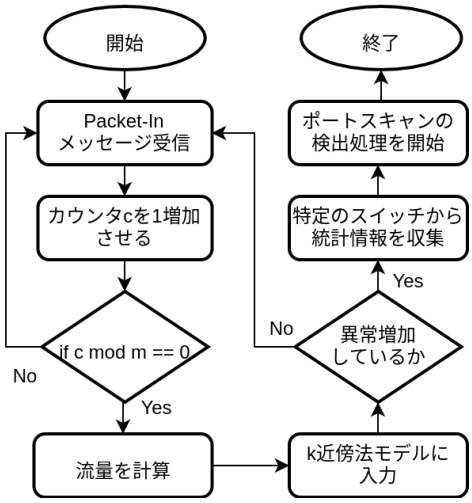

図 1: 提案手法のフローチャート

物理ポート番号（In-Port）が含まれている．本研究では，DPID と In-Port のセットを用いて，特定のリンク単位で Packet-In メッセージの流量をモニタリングする．このような手法を取ることで，ポートスキャンの検出処理時に統計情報を取得するスイッチを限定することが出来る．Packet-In メッセージの流量は単位時間当たりの Packet-In メッセージの数で表され，Packet-In メッセージを m 個受信する毎に式 (1) で求められ，記録される．

$\Delta Time$ は，現在の時刻と，前回に流量を計算した時刻の差を表す．

$$\text{流量} : V_p = \frac{m}{\Delta Time} \tag{1}$$

3.3 Packet-In メッセージ異常増加の検出手法

本研究では，記録された Packet-In メッセージの流量を元に，k 近傍法を用いて異常増加を検出する．具体的な処理手順を以下に示す．

Step :1 流量のデータ列に対してウィンドウサイズ W を設定する
Step :2 新しいデータが得られた場合，W 内の全てのデータとのユークリッド距離を計算する
Step :3 Step2 で計算した距離について，最も短い k 個のデータを選び，その平均値を新しいデータの異常スコアとする
Step :4 閾値を超えていた場合は異常と判定する

4 実験

提案手法について，ポートスキャンと Packet-In メッセージの流量との関係や，k 近傍法による異常検出の効果を確認するための実験を行った．実験のために，Ryu コントローラと Open vSwitch を用いて仮想的な OpenFlow ネットワークを構築した．

OpenFlow ネットワーク上のあるホストから正常なトラフィックを流し，その最中に当該ホストによるポートスキャンを発生させた．今回の実験で用いた各種パラメータを表 1 に示す．実験結果として，Packet-In メッセージの流量と経過時間の関係を図 2 に示す．

正常なトラフィックによる Packet-In メッセージ

表 1: 実験で用いたパラメータ

パラメータ	値	備考
m	20	流量の計算処理を行う間隔
W	20	k 近傍法のウィンドウサイズ
k	1	k 近傍法で用いる近傍数

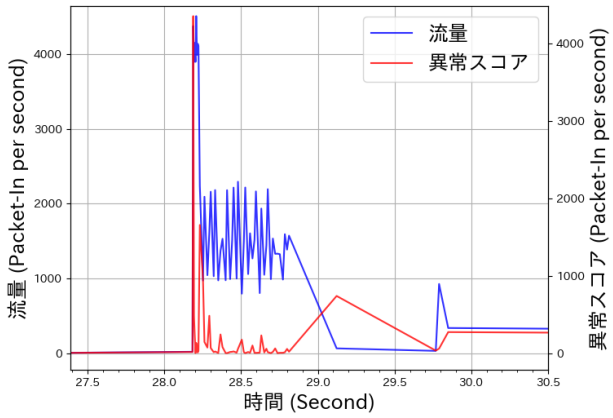

図 2: 実験結果

の流量は 2~3 pps (Packet-In per second) 程度であったが，約 28.2 秒の時点でポートスキャンが発生した際には約 4400 pps 規模の流量となっていることが分かる．k 近傍法を用いた際の異常スコアはポートスキャンの発生直後にピークを迎え，その後は比較的低い値で落ち着いている．従って，異常スコアの増加に対して適切な閾値を設定することで，ポートスキャン発生の推定が可能であることが確認できた．

5 おわりに

本稿では，Packet-In メッセージの流量からポートスキャンの発生を推定する手法について検討を行った．実験により，ポートスキャンが Packet-In メッセージの流量増加をもたらしており，k 近傍法を用いて発生の推定が可能であることが確認できた．今後は，データの振動に対して堅牢な異常検出アルゴリズムの考案と，異常を検出した後のポートスキャン検出処理，ホストの遮断処理に関して，実装および評価を行う．

参考文献

[1] C. V. Neu et al.: “Lightweight IPS for port scan in OpenFlow SDN networks,” Proc. of NOMS 2018, pp. 1-6. 2018.
[2] C. Yunhe et al.: “SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks,” Journal of Network and Computer Applications, vol. 68, pp.65-79, 2016.
[3] E. Bou-Harb et al.: “Cyber Scanning: A Comprehensive Survey,” IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1496-1519, 2014.