OpenFlowを用いた端末非依存型検知方式における負荷分散法

宮崎, 亮輔
九州大学大学院システム情報科学府情報学専攻

川本, 淳平
九州大学大学院システム情報科学研究院 : 助教

松本, 晋一
九州大学大学院システム情報科学府情報学専攻 | 九州先端科学技術研究所

櫻井, 幸一
九州大学大学院システム情報科学研究院 : 教授

出版情報：電気・情報関係学会九州支部連合大会. 平成27年度(第68回), pp.09-2P-07-, 2015-09-26
バージョン：
権利関係：

OpenFlow を用いた端末非依存型検知方式における負荷分散法

宮崎 亮輔 * 川本 淳平 * 松本 晋一 * 櫻井 幸一 *
(*九州大学 大学院 システム情報科学府 情報学専攻)

1 はじめに

近年，コンピュータネットワークは急速に普及してきている．それに伴い，以前は愉快犯によるサイバー攻撃が主であったのが，近年では愉快犯に加えて金銭や個人情報を狙ったサイバー攻撃へと，攻撃形態が多様化してきている．その一方で，ウイルス対策ソフト等を積極的に導入しないような，ネットワークセキュリティに対する意識が低いユーザが存在する．そういったユーザに対しても有効なセキュリティを提供することが必要である．

マルウェアから自身を守る手段として，従来様々な手法が確立されてきたが [1]，近年注目され始めた手法として，Moving Target Defense（MTD）という防衛概念 [2][3] が提案されている．しかし，MTD は常にユーザの環境が変化するので，正当なユーザにまで影響を及ぼしてしまうという問題が存在する．

この検知・防衛手法の問題を解決した既存手法として，Ant-Based Cyber Defense(ABCD)[4][5][6] が提案されている．この手法では，アリに見立てたパケット（アントパケット）をマルチエージェントとして利用している．一定の条件を満たした場合，アントパケットはフェロモンを落としながらネットワーク上を移動する．アントパケットがフェロモン濃度の高い経路に次第に集まって来ることで，脅威を検知する仕組みである．図 1 は ABCD の仕組みをまとめたものである．しかし，この手法では，防衛対象の各端末にソフトウェアの導入を必須としているので，ネットワークに接続する機能を持った組み込み機器に対して適用することが出来ないという問題が存在する．

そこで，我々は MTD を用いた攻撃検知を，OpenFlow を用いることで機器を改変することなく導入する手法 [7][8] を提案した．しかし，この手法では単一の OpenFlow コントローラがネットワーク全体を制御しているので，大規模なネットワークに対してはパケットドロップやネットワーク遅延等が発生するという問題が存在する．

本研究では，我々が提案した既存手法に対し，複数の OpenFlow コントローラを用意することで，適用可能なネットワークの範囲を広げる手法を提案する．各々のネットワーク毎に OpenFlow コントローラを用いることで，パケットドロップ率が改善されることを確認した．

2 提案手法

ネットワークを制御するコントローラを複数用意することで，従来単一のコントローラに集中していた負荷を分散させる．具体的には，ネットワーク上に存在する OpenFlow スイッチを任意の個数に分け，各々のグループに 1 台ずつコントローラを用意し，各コントローラが担当範囲内のスイッチを制御する．通常 OpenFlow スイッチを複数利用する場合は，各々のコントローラ同士の連携が必要である．それに対し，我々が提案した手法 [7][8] では，コントローラ同士の通信は一切不要であり，各々が独立して動作する．図 2 は，6 台の OpenFlow スイッチを 3 台毎に分け，各々のグループをコントローラ 1 及びコントローラ 2 の 2 台で分

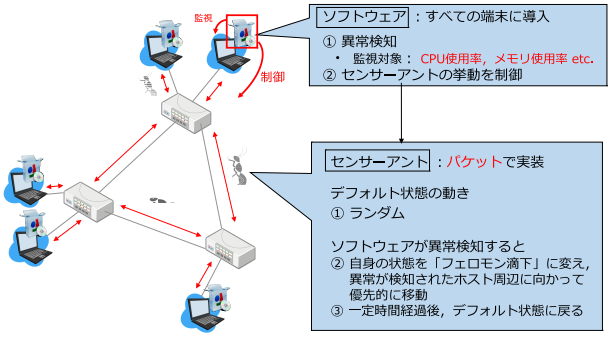

図 1: Ant-Based Cyber Defense の概念図

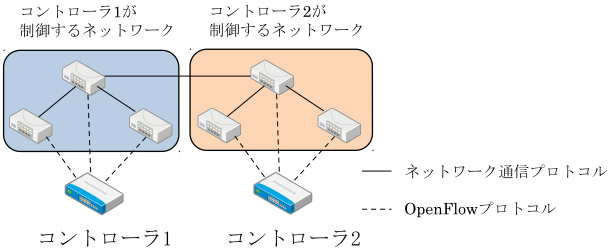

図 2: 複数コントローラを用いた負荷分散イメージ図

散制御する例を表している．ここで，実線は実際のネットワーク通信におけるプロトコルを表し，破線は OpenFlow プロトコルを表している．図 2 における実線部分，即ちスイッチ間をアントパケットが飛び回っている．このアントパケットは UDP パケットで実装されている．TCP プロトコルは再送要求を行うので，DoS 攻撃などが来た場合はアントパケットがネットワーク内に飽和してしまう．その負荷を軽減する為に UDP プロトコルを用いている．

3 実験

図 3 に示すツリー型ネットワークにおいて，ホスト 1〜ホスト 16 に対して，大学研究室内でキャプチャした正常通信データを流した．また，同時に CCCDATAset2008[9] によるボットネット通信データをホスト 5 に流し，ポートスキャンをホスト 13 に対して行った．パケットデータの再放流には Tcpreplay を用いた．この時，用いるコントローラを 1 台，2 台，5 台，10 台，15 台と変化させ合計で 5 回分の実験を行い，それぞれにおけるパケットドロップ率 R_{drop} を測定した．ここで，パケットドロップ率とは，生起したアントパケット数を A_{born}，途中でドロップせずにシステムによって正しく消滅したアントパケット数を A_{delete} とした時に，

$$R_{drop} = 1 - \frac{A_{delete}}{A_{born}} \quad (1)$$

によって表される数値である．アントパケットは s[秒] に 1 匹，各々のスイッチから発せられており，設定した寿命

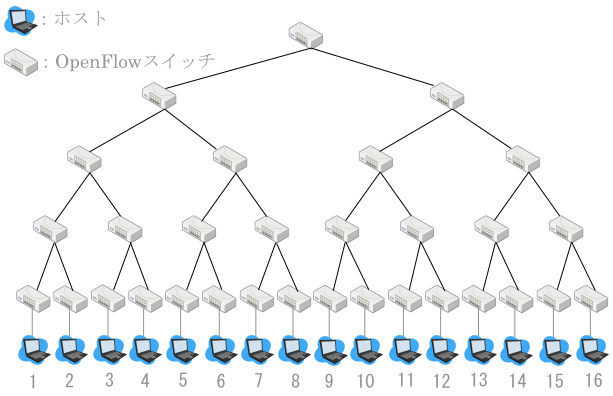

図 3: 実験で用いたツリー型ネットワーク

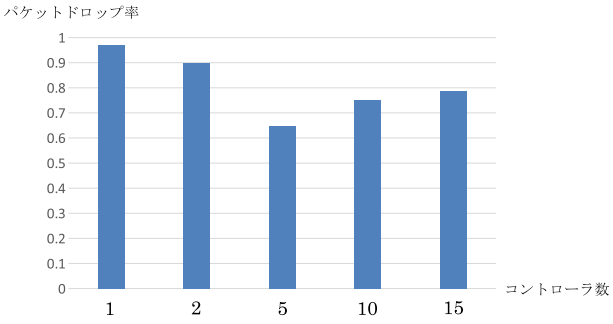

図 4: コントローラ数に対するパケットドロップ率の変化

に達するか巣に戻った場合はシステムによって自然に消滅する．ネットワーク全体のスイッチ数を n，本手法を運用した時間を t[秒] とした時，A_{born} は

$$A_{born} = \frac{nt}{s} \tag{2}$$

によって与えられる．この時，パケットドロップが発生しない理想的な場合は，$A_{born} = A_{delete}$ となる筈であり，この時のパケットドロップ率は 0 である．実際の実験では A_{delete} の数を測定することで，R_{drop} を算出した．また，$s = 1$，$n = 31$，$t = 900$ とした．

実験に用いたパラメータ及び A_{delete} の測定値から式 (1)(2) を用いて求めたパケットドロップ率のグラフを図 4 に示す．結果としては，コントローラ数を 1 台，2 台，5 台と増やしていくにつれ，パケットドロップ率は減少していった．しかし，コントローラ数が 10 台以降は，逆にドロップ率が増加してしまった．また，コントローラを n 台用いてもパケットドロップ率は $1/n$ 倍とはならなかった．これは，実験を全て同一の物理マシン内で行っており，コントローラを複数台用意しても，全体で見れば 1 つの同一プロセッサによって処理していることに起因していると思われる．コントローラ数が 5 台程度までは，パケットの処理を分散させることによる高速化の恩恵を受ける．しかしながら，10 台より増やしてしまうと，物理ホストマシンのプロセッサのリソースを共有することによるオーバーヘッドが，分散処理による高速化を上回ってしまい，逆に遅くなってしまうという結果となった．

4 まとめ

ネットワークの規模が大きな場合に，用いるコントローラの数を増やし処理を分散させることで，我々が従来提案していた手法におけるパケットドロップ率を改善した．その際に，コントローラの数を 1 台，2 台，5 台と増やしていくにつれパケットドロップ率は低下したが，10 台以降はパケットドロップ率が増加した．遅延対策に関しては，仮想環境で実験を行うのではなく，物理機器によるスイッチ，コントローラを用意し，リソース共有のオーバーヘッドを無くすだけでなく，実際の通信によるオーバーヘッドを考慮した実験を行う必要がある．また，今回の提案手法では，コントローラ間では何の連携もしておらず，各々が完全に独立して動作している．コントローラ間でパケットドロップ率や各々が担当するネットワーク内の情報等を共有し，より効率的な負荷分散アルゴリズムを考案することで，完全独立型と比較することも今後の課題である．

参考文献

[1] Manuel Egele, Theodoor Scholte, Engin Kirda, Christopher Kruegel , “A survey on automated dynamic malware-analysis techniques and tools” *ACM Computing Surveys (CSUR)*. Vol.44 Issue 2, 2012.

[2] Sushil Jajodia, Anup K. Ghosh, Vipin Swarup, Cliff Wang, X. Sean Wang, “Moving Target Defense” *Advances in Information Security*. Vol. 54, 2011.

[3] Jajodia, S., Ghosh, A.K., Subrahmanian, V.S., Swarup, V., Wang, C., Wang, X.S., “Moving Target Defense II” *Advances in Information Security*. Vol. 100, 2013.

[4] Jereme N. Haack, Glenn A. Fink, Wendy M. Maiden, A. David McKinnon, Steven J. Templeton, Errin W. Fulp, “Ant-Based Cyber Security” *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*. pp.918-926, 2011.

[5] Glenn A. Fink, Jereme N. Haack, A. David McKinnon, Errin W. Fulp, “Defense on the Move: Ant-Based Cyber Defense” *Security & Privacy, IEEE*. Vol. 12, Issue: 2, pp.36-43, 2014.

[6] Glenn A. Fink, A. David McKinnon, “Effects of Network Delays on Swarming in a Multi-agent Security System” *First International Workshop on Agents and CyberSecurity*. Article No. 11, 2014.

[7] 宮崎 亮輔, 川本 淳平, 松本 晋一, 櫻井 幸一, “攻撃検知のための端末非依存型システムを実現する OpenFlow コントローラの実装と評価” 火の国情報シンポジウム, 2015.

[8] 宮崎 亮輔, 川本 淳平, 松本 晋一, 櫻井 幸一, “ネットワーク攻撃に対する端末非依存型検知方式の OpenFlow コントローラ上への実装と評価” DICOMO2015.

[9] 畑田充弘, 中津留勇, 寺田真敏, 篠田陽一, “マルウェア対策のための研究用データセットとワークショップを通じた研究成果の共有” 情報処理学会シンポジウムシリーズ, Vol.2009, No.11, CSS2009(MWS2009), pp.1-8, 2009.