反射型 DDoS 攻撃に対する分散協調型の検知・防御

川頭 龍心*, 松本 倫子（九州産業大学）

Cooperative Decentralized Detection and Prevention of Distributed Reflective DoS Attacks

Ryushin Kawazu, Noriko Matsumoto (Kyushu Sangyo University)

1. はじめに

ネットワークへのサイバー攻撃は、社会的脅威にもなってきており、火急の対策が求められている。様々な攻撃がある内でも、DDoS 攻撃はサービスの低下や停止を引き起こし、深刻な影響を及ぼす。特に反射型 DDoS 攻撃は検知・防御が難しく、対応に工夫を要する。また、攻撃が大規模化・広域化していることから、もはやサーバ単体や組織のファイアウォールだけでなく、ネットワーク全体で対応していくことが望まれる。そこで本研究では、反射型DDoS攻撃の検知・防御について、ネットワーク機器にその機能を持たせ、さらにそれら複数を集中管理なしに分散協調的に連携させる方式を提案する。

2. DoS 攻撃とその変種

DoS（Denial of Service）攻撃は、特定のサーバ、ホスト、ネットワーク機器に大量のパケットを短時間に集中して送りつけ、機能不全に陥れる攻撃である。その変種として DDoS（Distributed DoS）攻撃、さらには反射型 DDoS 攻撃（Distributed Reflective DoS 攻撃とも呼ばれる）がある。DoS 攻撃の送信源は単一だが、DDoS 攻撃ではネットワーク上の多数の乗っ取られたホスト、反射型 DDoS 攻撃では多数の公開サーバである [1]。

反射型DDoS攻撃の攻撃者は、送信元IPアドレスを標的ホストの IP アドレスで詐称した要求パケットを、DNS、NTP、SNMP、memcached など、公開サーバの多数に送信する。それらのサーバは正規の応答パケットを標的ホストに送りつけ、結果として標的ホストは、大量のメッセージが短時間に集中して届くことから、機能不全に陥る。応答パケットそのものは正規なこともあり、攻撃の検知・防御が難しい。特に DNS を利用するものは、DNS リフレクタ攻撃などとも呼ばれる [2]。

3. 攻撃検知・防御の分散協調化

ネットワーク上の攻撃はますます大規模かつ広域になってきており、各組織のネットワークの入口などに侵入検知防御システム（Intrusion Detection and Prevention System, IDPS）を設置するだけでは対応に限界がある。それに加えて、ネットワーク内の各所にも IDPS を散在させて配置し、かつそれらを連携させることで、負荷分散や担当エリア分散を図るべきである。このような連携を集中管理するのは、負荷分散や即応性の観点から得策でなく、相互の分散協調で実現するのが望ましい。

このような複数 IDPS の分散協調は、別の観点からも有用である。攻撃の検知・防御には様々なパラメータがあり、誤検知をできるだけ避けるための最適なパラメータ設定は必ずしも不変かつ既知ではなく、状況によっても変化する。そこでパラメータ設定の異なる複数の IDPS を同時に運用して相互に情報交換など連携させることで、互いに補填しあって最適なものに近づけていくことが期待できる。同様の考え方は、例えば機械学習の分野でもアンサンブル学習として活用されている。

4. 本研究の目的

我々は、以上のような背景や関連技術動向を踏まえ、侵入検知・防御を複数のネットワーク機器に担わせて連携させ、ネットワーク上に仮想的な IDPS を実現することを目指している。実装プラットフォームとして Software Defined Networking（SDN）の OpenFlow を採用し、まず Portscan 攻撃について、検知・防御をネットワーク上で分散協調化する研究を進め、成果を挙げた [3]。SDN による IDPS について、複数連携や分散協調に関する研究事例は、調べた限りでは他に見当たらない。

本研究は、同様の手法を反射型 DDoS 攻撃の検知・防御に展開し、実現可能性を探ることを目的としている。検知・防御に SDN を用いる関連研究として、DDoS 攻撃については Braga らの研究 [4]、反射型 DDoS 攻撃については首藤らの研究 [5]、吉田の研究 [6] がある。前者は DNS リフレクタ攻撃について、AS（Autonomous System）の入口で DNS 応答パケットを監視して検知・防御する方式で、特に、防御する間も標的ホストが DNS を参照できるような SDN の利用を提案しており、本研究とは目指す方向が異なる。後者は各 DNS サーバの近くで要求パケットを監視する方式を提案しており、本研究の下地の一部としている。

5. SDN と OpenFlow

SDN は、従来のルータにおける制御部（コントローラ）と転送部（スイッチ）を分離し、制御部の集約やソフトウェア化を可能にするアーキテクチャであり、OpenFlow はその実装仕様である。コントローラからスイッチへは転送指示など制御情報を渡し、スイッチからコントローラへは状況情報やパケット数など統計情報を渡す。なお、スイッチはパケットのヘッダは参照更新できるが、ペイロードは参照更新できない。OpenFlow を IDPS として利用する場合、通信内容に関わらない監視・防御を扱うことになる。

分散協調の連携を実現するには、近隣のコントローラどうしを接続することになる。その規格は定められていないが、可能である。コントローラどうしの接続を扱った関連研究は幾つかあるが、コントローラの多重化による障害対策がテーマとなっている。しかし、それらで用いられた接続方法は、本研究でも参考にした。

本研究では OpenFlow の実験テストベッドとして、ネットワーク・エミュレータ Mininet [7]、コントローラのフレームワーク Ryu [8]、仮想スイッチ Open vSwitch [9] を用いた。

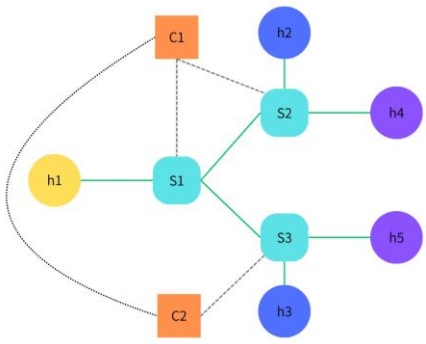
6. コントローラ連携による監視・防御

コントローラは基本的に次のような動作を繰り返す。

(1) DNS 要求パケットおよび応答パケットを一定間隔で計数。
(2) 数値が閾値を越えたら攻撃検知、越えなければ (1) に。
(3) 攻撃を検知したら応答パケット遮断、他コントローラに通知。
(4) 検知の通知が来たら、自らはまだでも、応答パケット遮断。
(5) (1) に戻って監視を継続。

攻撃検知については、上の (4) のように、コントローラごとの未然防御を可能にしている。また、パケット遮断については、一定のタイムアウト時間で解除されるので、それより少し短い時間ごとに監視を繰り返し、攻撃継続中なら遮断設定を更新して、攻撃継続を他コントローラに通知する。攻撃継続の通知を受け取ったコントローラは、自らの遮断設定を更新する。このようにコントローラ間では、標的ホストの IP アドレスおよび「新規攻撃検知」「攻撃継続」の 2 種類の通知を互いに授受し合う。

検知の主なパラメータとして、閾値とタイムアウト時間がある。閾値が小さすぎると誤検知の恐れが高まり、逆に大きすぎると手遅れの恐れが高まる。タイムアウト時間は複数コントローラ間でずらすことで、1 台よりも効果的な検知が期待できる。しかし、いずれも適切な値を事前に予測できないので、パラメータ値の異なる複数を連携させる意義がある。



h1: 標的ホスト　　C1, C2: コントローラ
h2, h3: DNS サーバ　　S1, S2, S3: スイッチ
h4, h5: 攻撃ホスト

図 1. 実験ネットワークの構成
Fig.1. Topology of the experiment network.

7. 実験システム

以上を具体化する実験システムは、最初の取組みなことから最小限の構成とし、コントローラ 2 台で、図 1 のように構成した。

h4 と h5 はそれぞれ h2 と h3 に、h1 を送信元と詐称した要求パケットを送り、h2 と h3 は h1 に向けて応答パケットを返す。その状況をスイッチ S2 と S3 および S1 で監視して、C1 と C2 は必要に応じてパケットを遮断する。C1 と C2 は互いに接続していて、それぞれの情報を交換する。このように、ネットワーク内に散在するスイッチの内、標的に近い S1 と遠い S2 との連携も想定している。また、閾値の違いの影響より先に、まずタイムアウト時間の違いの影響を知るべく、今回は C1 と C2 の閾値は同一、タイムアウト時間が異なる設定とした。

この実験システムをテスト・シナリオで動作させ、次のような挙動などを確認した [10]。

- 攻撃を先に検知したコントローラが他方に通知することで、未然防御ができていること。
- 特に、標的から遠いスイッチではまだ検知できなくても、近いスイッチで先に検知できること。
- タイムアウト時間を異なる設定とすることで、異なる時点での監視ができること。
- 攻撃が去った際には、全てのスイッチで防御の終了が自律的になされること。

8. おわりに

本研究では、反射型 DDoS 攻撃に対する分散協調型の検知・防御について、OpenFlow を活用して、まず最初の実現可能性は示せたと考えている。しかし、最小規模の実験で最低限の動作を確認したにすぎない。より大きなネットワークでの実験、より様々な攻撃パターンでの実験、より多数のコントローラの連携、より高度な複数パラメータ連携、さらには、OpenFlow を越えて、通信内容にまで踏み込んだ監視・防御の可能性の模索などに、引き続き取り組んでいく予定である。

謝辞 貴重なご助言を頂いた吉田紀彦先生（埼玉大学名誉教授・立正大学研究員）に深謝する。本研究は科研費 23K11071、25K15086 の助成を受けている。

文 献

(1) 臼﨑: 宮崎大学 学位論文 (2022)
(2) https://jprs.jp/glossary/index.php?ID=0156 (2025 閲覧)
(3) 中村, 他: 電気学会 全国大会論文集, 3, 65-66 (2022)
(4) R. Braga, et. al.: Proc. IEEE LCN, 408-415 (2010)
(5) 首藤, 他: 信学会 技術研究報告, 113(473), 223-228 (2014)
(6) 吉田: 埼玉大学 卒業論文 (2021)
(7) https://mininet.org/ (2025 閲覧)
(8) https://ryu-sdn.org/ (2025 閲覧)
(9) https://www.openvswitch.org/ (2025 閲覧)
(10) 川頭: 九州産業大学 卒業論文 (2026)