IDS・SDN連携型ファイアウォールシステムにおける IDS多重化による攻撃遮断高速化

並木 涼[1,†1]　サリチ エルトゥール[1,†2]　山井 成良[1,a)]

概要：ファイアウォールシステムの構成法として IDS（Intrusion Detection System）と SDN（Software Defined Network）システムとを併用する IDS・SDN 連携型ファイアウォールシステムが構成や機能の柔軟性の観点から注目されている．この構成法では IDS に全てのパケットが複製・転送されるため，IDS の負荷増大が問題となる．本稿では IDS の多重化により IDS の負荷分散を図る方法を提案する．これにより従来の構成に比べて高速な遮断動作が期待できる．

1. はじめに

ファイアウォールは多くの場合組織内ネットワークと組織外ネットワークとの境界に設置され，主に組織外ネットワークからの攻撃や不正なアクセスを解析してこれらを遮断し，安全な通信のみを許可するシステムを指す [*1]．ネットワーク経由でのサイバー攻撃が年々増加している現代社会では，ファイアウォールは必要不可欠な存在であると言える．

ファイアウォールには IP アドレスやポート番号などのレイヤ 2-4 情報に基づくフィルタリングを行うものや，DPI（Deep Packet Inspection）のようにペイロードの検査を行ってその結果に基づきフィルタリングを行うものなどが含まれており，様々な構成法がある．その 1 つとして，不正通信を IDS（Intrusion Detection System）により検出し，これを SDN（Software Defined Network）により遮断する構成法（以下，IDS・SDN 連携型ファイアウォールシステム）[1] が知られている．この構成法ではたとえば検査内容の異なる IDS を用いて不正通信の検出精度を高めたり，SDN で単に不正通信を遮断するだけでなくハニーポットに誘導したりすることも可能であり，従来の構成法より構成や機能の柔軟性が高い点で注目されている．

[1] 東京農工大学
Tokyo University of Agriculture and Technology
2-24-16, Nakacho, Koganei, Tokyo 184–8588, Japan
[†1] 現在，SATORI 株式会社
Presently with SATORI, Inc.
[†2] 現在，日本 IBM 株式会社
Presently with IBM Japan, Ltd.
[a)] nyamai@cc.tuat.ac.jp
[*1] ファイアウォールの定義はまちまちであるが，本稿では不正通信を遮断するシステム全般を指し，IPS（Intrusion Protection System）の機能を含むものとする．

IDS・SDN 連携型ファイアウォールシステムでは IDS から SDN コントローラへの遮断内容の通知方法が設計上の問題として重要となる．桂らは IDS からのアラートメッセージをログファイルに記録し，ログ監視ツールを用いてこれを検出して REST（Representational State Transfer）[2] により SDN コントローラに通知する方法を用いた [3]．しかし，この方法ではログ監視のオーバヘッドが大きく，遮断動作が遅い点が問題となっていた．これに対して，我々は OpenFlow を用いた IDS・SDN 連携型ファイアウォールシステムにおいて IDS が syslog あるいは SNMP（Simple Network Management Protocol）トラップによりアラートメッセージを出力し，これを受け取った OpenFlow スイッチがそれを Packet-In により OpenFlow コントローラに中継し，これを OpenFlow コントローラが解析して OpenFlow スイッチに通信を遮断するようにフローエントリを追加する方法を提案した [4], [5]．

しかし，IDS・SDN 連携型ファイアウォールシステムでは双方向の全てのトラフィックが IDS に渡されるため，IDS の負荷が大きくなりやすく，高速な遮断動作を行う上でボトルネックになる懸念がある．そこで本稿では複数の IDS を導入して IDS 間で負荷分散を行い，遮断動作の更なる高速化を図る方法を提案する．

2. 関連研究

文献 [4], [5] では IDS・SDN 連携型ファイアウォールシステムにおいて syslog および SNMP トラップによりアラート通知を行う方法を提案した．以下では**図 1** に示すように SDN として OpenFlow を用いる場合のシステム構成において IDS から OpenFlow コントローラに syslog でアラー

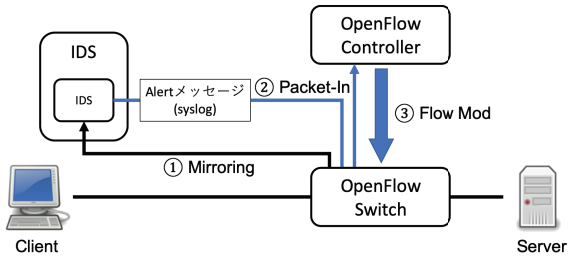

図 1 IDS から OpenFlow コントローラへの syslog を用いたアラート通知 [4]

ト通知を行う方法を説明する．

同図において IDS，OpenFlow スイッチ，OpenFlow コントローラは次のように動作する．

(1) OpenFlow スイッチはクライアント・サーバ間でやり取りされるパケットをミラーリングにより IDS に転送する．

(2) IDS は受信したパケットがアラート対象と判断した場合，アラートメッセージを syslog プロトコルにより UDP パケットとして OpenFlow スイッチに送信する．

(3) OpenFlow スイッチは受信したアラートメッセージを Packet-In により OpenFlow コントローラに転送する．

(4) OpenFlow コントローラは受信したアラートメッセージを解析し，制御対象となるトラフィックを特定する．

(5) OpenFlow コントローラは OpenFlow スイッチに対して制御対象となるトラフィックの遮断やハニーポットなどに転送するための設定（FlowMod）を行う．

アラートメッセージとして SNMP トラップを用いる場合でも，OpenFlow コントローラによるアラートメッセージの解析方法が異なるだけで，動作手順は同じである．なお，アラートメッセージの送信先 IP アドレスは実在する syslog サーバあるいは SNMP マネージャである必要はないが，これらが実在すれば OpenFlow スイッチでは OpenFlow コントローラへの Packet-In だけでなくこれらへの転送も行うようにフローエントリを設定しておく必要がある．

上記の動作により，IDS は syslog メッセージや SNMP トラップを出力できる任意のものを使用することができ，ログ監視ツールを用いる必要がないことからオーバヘッドを削減する効果が期待できる．プロトタイプシステムで性能評価実験を行った結果，OpenFlow スイッチがパケットをミラーリングしてからアラートメッセージを中継するまでの時間が従来のログ監視ツールを用いる方法では 700ms であったのに対して syslog を用いた方法では 527ms となり，173ms のオーバヘッド削減効果が確認された．また，実験環境が異なるため従来のログ監視ツールを用いる方法との直接的な比較は行っていないが，SNMP トラップを用いた方法では遮断動作が完了するまでの時間が 555.3ms となり，syslog を用いた方法の 566.1ms と比較して同等以上の性能を有することが確認できた．

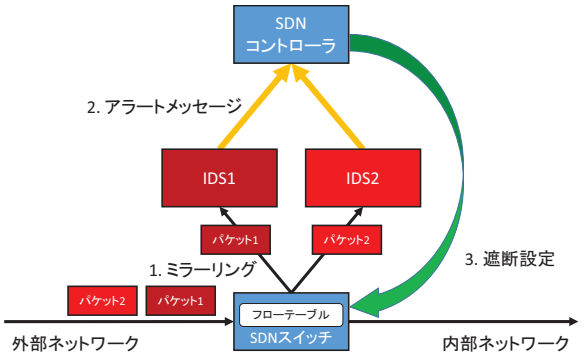

図 2 2 台の IDS を用いた IDS・SDN 連携型ファイアウォールシステムの構成例

3. IDS 多重化による攻撃遮断高速化

3.1 システム構成

前節で述べたように，IDS・SDN 連携型ファイアウォールシステムにおいて syslog メッセージや SNMP トラップを OpenFlow コントローラが直接解析する方法を導入することにより，従来のログ監視ツールを用いる方法と比べて遮断動作に要する時間の大幅な短縮が可能になった．しかし，それでも遮断動作に 550ms 以上の時間を要しており，更なる高速化が必要な状況である．そこで IDS 多重化による遮断動作の高速化を図ることにした．

IDS・SDN 連携型ファイアウォールシステムでは SDN コントローラは任意の IDS からのアラートメッセージを受け取って遮断動作を行えるため IDS の多重化は容易である．2 台の IDS を用いた構成例を**図 2** に示す．SDN スイッチに到着した外部ネットワークからのパケットは予め SDN スイッチに設定したフローテーブルに従っていずれかの IDS にミラーリングされる．各 IDS は独立して受信したパケットが不審であるかどうかを判定する．IDS がパケットを不審と判定した場合，SDN コントローラにアラートメッセージを送り [*2]，SDN コントローラは SDN スイッチに対して当該パケットによる攻撃を遮断するようにフローテーブルを変更する．

3.2 IDS 間での負荷分散

図 2 の構成例において，IDS 間でどのように負荷分散を行うかが設計上の重要な問題となり得る．もし IDS が 1 つのパケットだけで不審であるかどうかを判定できるのであれば，SDN スイッチはパケットをランダムに任意の IDS にミラーリングして転送しても問題ない．しかし，DPI のように一連のパケットに基づいて不審な通信かどうかを判定する場合には，同一フローに属するパケットを同一の ISD にミラーリングする必要がある．

IDS 間でこのような負荷分散を実現する方法の 1 つと

[*2] 実際には SDN コントローラに直接送るのではなく，SDN スイッチから Packet-In により送られる．

して，ポート番号に基づく振分けが考えられる．たとえば電子メールのプロトコルである SMTP（25/TCP）や，WWW のプロトコルである HTTP（80/TCP）を使用する通信を専用の IDS にミラーリングする方法が考えられる．しかし，特にこれらのプロトコルを使用する通信とその他の通信が均等に発生する状況でなければ負荷分散の効果が限定的となる．一方，たとえば外部ホストの IP アドレスの最下位ビットが 0 か 1 かに応じて 2 台の IDS に振り分ける方法であればほぼ均等に負荷分散を行うことが可能になる．

このような負荷分散は市販の負荷分散装置を用いれば容易に実現できる．しかし，このような装置は一般的に高価であるため，本稿では SDN スイッチで負荷分散機能を実現するようにする方法を提案する．たとえば，2 台の IDS を用いる場合において，外部ネットワーク側の IP アドレスに対してネットマスク 0.0.0.1 を適用した結果 0.0.0.0 であれば IDS0 に，0.0.0.1 であれば IDS1 にミラーリングするようにすれば IDS0, IDS1 間でほぼ均等に負荷分散する事が可能になる．用意できる IDS の台数によっては必ずしも各 IDS 間で均等に負荷分散を行えないが，たとえば IDS が 3 台の場合，外部ネットワーク側の IP アドレスに対してネットマスク 0.0.0.7 や 0.0.0.15 などを適用し，3 で割った余りに基づいて異なる IDS に振り分ける方法が考えられる．

3.3 試作システムの実装

我々は従来の IDS・SDN 連携型ファイアウォールシステムを改変し，2 台の IDS を用いた IDS・SDN 連携型ファイアウォールシステムを試作した．試作システムの構成 [*3] を**図 3** に示す．この図において OpenFlow スイッチとして動作する Open vSwitch および OpenFlow コントローラとして動作する Ryu は 1 台の Raspberry Pi 4 に同居させ，IDS（Snort），クライアント，サーバはそれぞれ Raspberry Pi 3 を用いた．なお，IDS から OpenFlow コントローラへの通知方法は [3] と同様にログ監視ツール（swatch）が REST API（curl）を使用する方法とした．

Open vSwitch のフローテーブルの初期設定を**表 1** に示す．この設定により，前節で述べたようにクライアントの IP アドレスに基づいてほぼ均等に 2 台の IDS に負荷分散することが可能になる．

また，主な構成要素の諸元を**表 2** に，Open vSwitch, Ryu，Snort のバージョンを**表 3** に示す．

3.4 試作システムの性能評価

次に試作システムにおいて IDS 多重化による遮断動作時間の短縮効果を確認するため，性能評価実験を行った．

*3 簡単化のため 2 台目の IDS は省略している．

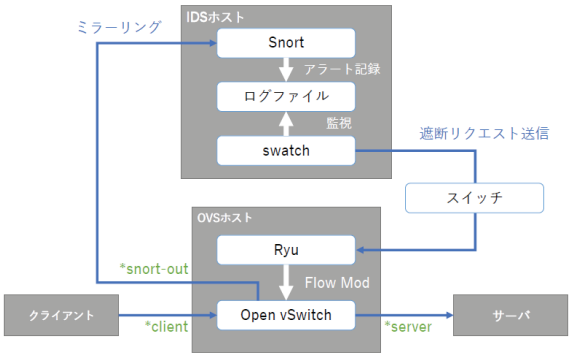

図 3 試作システムの構成

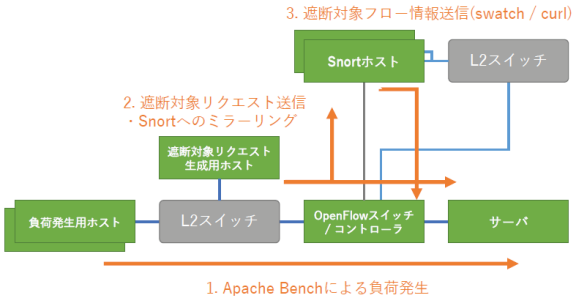

図 4 性能評価実験における実験環境

```
alert tcp any any -> any 80 (msg: "Attack detected";
content: "attack"; sid:1000002;)
```

図 5 IDS における snort の攻撃検出ルール

図 4 に実験環境を示す．ここで 2 台の負荷発生用ホストおよび遮断対象リクエスト生成用ホストはいずれも表 2 におけるクライアントと同じ諸元を持つ．

この実験環境において，2 台の負荷発生用ホストから Apache Bench（ab コマンド）を用いて 50 並列（-c 50）でペイロードサイズが 9.8kB, 98kB, 489kB, 970kB[*4] の大きさの HTTP POST リクエストを連続して送出するようにし，一定時間経過後に遮断対象リクエスト生成用ホストから「attack」の文字列を含む HTTP POST リクエストを送出させ，OpenFlow スイッチに遮断対象の HTTP リクエストが届いてから IDS による OpenFlow コントローラへの遮断リクエストが到着するまでに要する時間を Wireshark で計測し，これを遮断時間とした．IDS では snort に**図 5** に示す攻撃検出ルールを設定した．なお，システムを構成する Raspberry Pi のクロック周波数は最大値に固定とした．具体的には Raspberry Pi 4 では 1500MHz に，Raspberry Pi 3 では 1400MHz で動作するように設定した．ペイロードサイズと IDS の台数の 2 つのパラメータの組合せ（計 8 通り）に対して，それぞれ 10 回の測定を行い，遮断時間を記録した．

ペイロードサイズと IDS の台数の組合せに対する遮断

*4 パケットサイズの合計はそれぞれ 10kB, 100kB, 500kB, 1MB

表 1 Open vSwitch におけるフローテーブルの初期状態

優先度	マッチ条件	適用アクション
1	in_port=client, src_address=0.0.0.0/0.0.0.1	out_port=server,snort-out0
1	in_port=server, dst_address=0.0.0.0/0.0.0.1	out_port=client,snort-out0
1	in_port=client, src_address=0.0.0.1/0.0.0.1	out_port=server,snort-out1
1	in_port=server, dst_address=0.0.0.1/0.0.0.1	out_port=client,snort-out1
0	in_port=client	out_port=server
0	in_port=server	out_port=client

表 2 主な構成要素の諸元

役割	モデル名	プロセッサ	クロック周波数	メモリ容量	ネットワーク帯域
OpenFlow スイッチ兼コントローラ	Raspberry Pi 4 Model B	BCM2711	1.5GHz	4GB	1Gbps
IDS，クライアント，サーバ	Raspberry Pi 3 Model B+	BCM2837B0	1.4GHz	1GB	300Mbps

表 3 主な構成要素で使用するソフトウェア

役割	ソフトウェア名，バージョン
OpenFlow スイッチ	Open vSwitch version 2.10.1
OpenFlow コントローラ	Ryu version 4.34
IDS	Snort version 2.9.17
クライアント	Apache Bench version 2.3
サーバ	nginx version 1.14.2

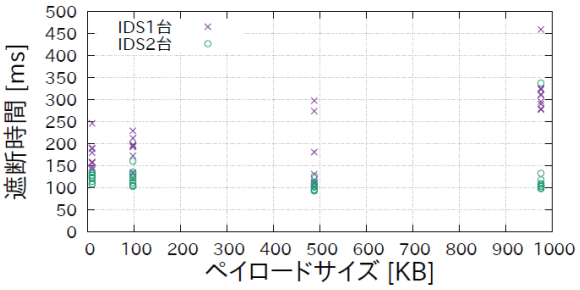

図 6 ペイロードサイズと IDS 台数の組合せに対する遮断時間の分布

時間の測定結果を**図 6** に示す．また，測定結果の中央値ならびに四分位範囲を**表 4** に示す．これらの図表からわかるように，遮断時間には多少の外れ値を含むものの，IDS が 1 台の場合はペイロードサイズに応じて 100ms から 190ms∼330ms 程度までの範囲，IDS が 2 台の場合は 90ms から 140ms 程度までの範囲に収まっていることがわかる．また，四分位範囲は IDS が 2 台の場合は IDS が 1 台の場合の 21∼44%となっており，遮断時間が安定して短縮されていることが確認できる．これらの結果から IDS の負荷分散が遮断時間の短縮に有効であるといえる．

4. まとめ

本稿では IDS・SDN 連携型ファイアウォールシステムにおいて攻撃遮断時間を短縮するために IDS を多重化する方法を提案した．また，性能評価実験により，IDS 多重化が攻撃遮断時間の短縮に有効に機能することを確認した．今後の課題としては，文献 [4], [5] で述べた，アラートメッセージを直接 SDN コントローラが解析して遮断する方法と組み合わせ，攻撃遮断時間の更なる短縮を図ることが挙げられる．

表 4 遮断時間の中央値と四分位範囲

(a) 中央値 [ms]

IDS 台数	ペイロードサイズ			
	9.8KB	98KB	489KB	970KB
1	158.1	193.7	115.5	311.4
2	123.7	119.9	102.2	105.9

(b) 四分位範囲 [ms]

IDS 台数	ペイロードサイズ			
	9.8KB	98KB	489KB	970KB
1	37.2	53.1	57.9	32.9
2	12.2	23.2	12.1	12.7

参考文献

[1] Paul Zanna, Benjamin O'Neill, Pj Radcliffe, Sepehr Hosseini, MD. Salman Ul Hoque: "Adaptive Threat Management Through the Integration of IDS Into Software Defined Networks", *Proceedings of 2014 International Conference and Workshop on the Network of the Future (NOF2014)*, pp.13–17, December 2014.

[2] Roy Thomas Fielding: "Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST)" (online), available from ⟨https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm⟩ (accessed 2021-05-10), 2000.

[3] 桂祐成，君山博之，堤智昭，米﨑直樹，丸山充: "ソフトウェアスイッチを使ったリアルタイム総当たり攻撃検出遮断システムの提案"，電子情報通信学会技術研究報告，NS2018-272, pp.461–464，2019 年 3 月．

[4] 桂祐成，児玉伊太郎，Pranpariya Sakarin，山井成良，君山博之，Vasaka Visoottiviseth : "IDS・SDN 連携型ファイアウォールシステムにおける遮断動作の高速化"，インターネットと運用技術シンポジウム 2019 論文集，Vol.2019，pp.116–117，2019 年 12 月．

[5] エルトゥールサリチ，並木涼，山井成良 : "IDS・SDN 連携型ファイアウォールシステムにおける SNMP を用いたアラート通知"，情報処理学会インターネットと運用技術研究会研究報告，Vol.2021-IOT-53/2021-CSEC-93，No.10，pp.1–4，2021 年 5 月．