Vol.2012-DPS-150 No.17
Vol.2012-CSEC-56 No.17
2012/2/29

IPトレースバックフローサンプリング手法における実攻撃を想定したハッシュ値生成手法の提案

瀬尾 奨太[†1] 櫨山 寛章[†1] 垣内 正年[†1]
猪俣 敦夫[†1] 藤川 和利[†1]

本論文ではこれまでの単一パケット追跡を行うIPトレースバック・ダイジェスト手法に対して，DDoS攻撃をフローで追跡するフローダイジェスト方式を提案する．具体的には，DoS攻撃はASを越えて行われるため，攻撃経路上にフルキャプチャ，sFlow，NetFlowのパケット収集手法が混在する環境でもトレースバックを継続できるように，いずれの手法でも収集可能な値を用いて，DDoS攻撃の主な攻撃手法であるフラッドベース攻撃，プロトコルベース攻撃，アプリケーションベース攻撃，それぞれの攻撃毎の特徴を抽出した値であるフローダイジェストを定義した．本論文ではフローダイジェスト方式をツリー構造トポロジーを用いてエミュレーション実験を行い，パケットを収集する割合（サンプリングレート）を一定に抑えた中での追跡を実現した．

Proposal of flow digest approach on Hash-based IP traceback against actual attacks

Shota Seo,[†1] Hiroaki Hazeyama,[†1]
Masatoshi Kakiuchi,[†1] Atsuo Inomata[†1]
and Kazutoshi Fujikawa[†1]

In this paper, we define flow digest approach on hash-based IP traceback to detect three actual DDoS attacks, application based attack, protocol based attack and flood based attack. Flow digest approach uses such values of a packet as hash inputs that can be collected by sFlow, NetFlow and full- capturing device respectively. By defining a flow of DDoS attack with using values of IP packet header, it can achieve higher traceability without raising sampling rate of collecting packets. We define flow digest values and validate our flow digest approach by emulation experiments in a tree-topology.

†1 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

1. はじめに

攻撃元を詐称したDDoS攻撃の対策の1つにIPトレースバックがある．ぞのうち，AS（Autonomous System）を越えて行われるDoS攻撃に対してAS間で問い合わせを行い，円滑に追跡を行う仕組みをAS間トレースバックと呼ぶ．IPトレースバック手法のうち，パケットの特徴を抽出した値であるパケットダイジェストとそのハッシュ値を用いて追跡を行うダイジェスト手法は，AS内の情報を外に漏洩する危険性もなく，各ルータに実装する必要がなく，導入コストも低く抑えられるためAS間トレースバックに適した手法として実用化が期待されている．

ダイジェスト手法には通過する全てのパケットのハッシュ値を記録するフルキャプチャ方式と，ある一定の確率Rでパケットを収集して，そのハッシュ値を記録するサンプリング方式[1)]がある．フルキャプチャ方式を用いた場合，追跡成功率は高いもののハッシュ値保存におけるメモリ，パケット収集のための運営コストが掛かるため，全てのAS境界ルータ（ASBR）でフルキャプチャ方式を取り入れる事は難しい．サンプリング方式はパケット収集，保存のためのコストは削減できるものの，実用的な追跡成功率は実現できていないのが現状である．従って，AS間トレースバックを実現するためにはサンプリング方式の追跡成功率を向上させるシステムの提案が必要である．また，AS間トレースバックを実現するには，各ASがダイジェスト手法の中でフルキャプチャ方式，sFlow，NetFlowなどのサンプリング方式などの中から，予算やポリシーに応じて選択できる環境が好ましいため，どのパケット収集方式を用いていても追跡可能なシステム設計でなくてはならない．

本研究ではこれらの要件を踏まえ，実攻撃を想定したフロー追跡用ハッシュ値生成手法であるフローダイジェスト方式を提案する．具体的には，フルキャプチャ，sFlow，NetFlowのどのパケット収集方式でも収集可能なパケット情報で，DoS攻撃の代表的な攻撃手法であるフラッド，プロトコルベース，アプリケーションベースの3つの攻撃を想定したフロー追跡用のハッシュ値定義を行い，ツリー構造トポロジーを用いてフローダイジェスト方式の追跡性を検証する．

2. 関連研究

追跡成功率を向上させる方式として甲斐ら[2)]はパケット追跡を主眼とし，既存のダイジェスト手法であるSPIEの方式[3)]を基にハッシュ値生成手法の改良を行った．甲斐らは誤検知率を削減ためにSPIE方式のハッシュ値に使用する値（パケットダイジェスト）を，1つの通信毎に共通する値（D1）と，1つのパケット毎に異なる値（D2）の2つに分けて定義し，単一パケット追跡に加え，DoS攻撃のフロー追跡を行った．**図1**はSPIEと甲斐氏の方式におけるパケットダイジェストである．この手法はダイジェスト手法フルキャプチャ方式に

Vol.2012-DPS-150 No.17
Vol.2012-CSEC-56 No.17
2012/2/29

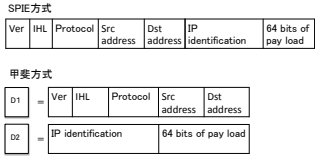

図 1 SPIE 方式と甲斐方式でのハッシュ値の定義

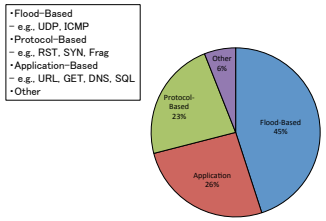

図 2 DDoS 攻撃被害報告の割合

表 1 各収集手法で取得可能な情報

	フルキャプチャ	sFlow サンプリング	NetFlow サンプリング
バージョン情報	○	○	○
パケット長	○	○	○
ヘッダ長	○		
送信元IPアドレス	○	○	○
宛先IPアドレス	○	○	○
TCPフラグ	○	○	○
IP識別子	○		
L4送信元ポート番号	○	○	○
L4宛先ポート番号	○	○	○
L4プロトコルタイプ 6=TCP, 17=UDP	○	○	○

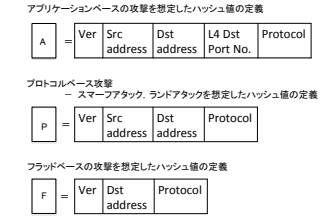

図 3 ハッシュ値の定義

おいてパケット追跡を主眼とした追跡率の向上を目的としている．本研究では複数の攻撃パケットに対し共通のハッシュ値を生成することで，攻撃経路上のエッジルータが追跡対象の攻撃パケットのハッシュ値を保持しやすくなるという点において，サンプリング手法の追跡成功率低下の問題解決に応用できるのではないかと考えた．

3. フローダイジェスト方式

本章ではフローダイジェスト方式の想定する実攻撃とトレースバックに用いるサンプリング装置を述べ，それを踏まえたハッシュ値の定義付けを行う．

3.1 想定する攻撃

図 2 は ArborNetworks 社による 2010 年までに報告された DoS 攻撃の分類とその割合[4]を元に，フローダイジェスト方式を定義する上で必要となるプロトコルヘッダの内容から再分類したグラフである．本研究では DoS 攻撃を以下のように分類し，フローの定義を行った．

- フラッドベース攻撃：接続を確立せずに大量に攻撃パケットを送りつける攻撃（- UDP Flood, ICMP Flood, RST Flood）
  - 送信元 IP アドレスは断続的に変化する恐れがある．
  - レイヤ 4 の宛先ポート番号は変化する．
  - レイヤ 4 プロトコルタイプは固定される．
- プロトコルベースの攻撃：送信元 IP アドレスを攻撃対象者と同じものに偽装し，他のサーバもしくは攻撃対象ノードへ問い合わせパケットを送信し，その返答を攻撃対象者へ誘導する攻撃（- Land attack, DNS Reflection）
  - 送信元 IP アドレスは偽装されるが断続的に変化することはなく固定される．
  - レイヤ 4 プロトコルタイプは固定される．
- アプリケーションベースの攻撃：アプリケーションベースで大容量，もしくは大量のパケットを攻撃対象者へ送りつける攻撃（- URL, GET, SQL）
  - TCP 接続を確立するため，送信元 IP アドレスは固定される．
  - 攻撃によってアプリケーションが固定されるので，レイヤ 4 の宛先ポート番号の値は固定される．
  - TCP 接続を確立するためレイヤ 4 プロトコルタイプは 6 で固定される．

本論文ではフロー追跡のためにパケットの特徴を元に分類し直したフラッドベース攻撃，プロトコルベースの攻撃，アプリケーションベースの攻撃の 3 種類の攻撃を想定してフローダイジェストの定義を行っていく．

3.2 パケットサンプリング装置

Blanc らの研究[5] では，単一 AS 内での IP トレースバック技術として sFlow， NetFlow が利用可能か調査した．その結果，たとえ送信元 IP アドレスの偽装されているパケットであっても，BGP 拡張をつけた sFlow パケットを解析することにより，追跡対象パケットの BGP 経路上のどの隣接 AS から流入，流出したかを追跡可能であることがわかった．本研究では sFlow，NetFlow 及びフルチャプチャによる方法でパケット情報の収集を行う．本研究では各 AS がフルキャプチャ，sFow，NetFlow どの手法を用いていてもトレースバックの継続ができるように，すべての手法で取得可能な値を用いてハッシュ値の定義付けを行う．フルキャプチャ，sFlow，NetFlow で取得可能な値を**表 1** に示す．

3.3 フローダイジェスト方式のハッシュ値の定義

本提案では, アプリケーションベース攻撃用, プロトコルベース攻撃用, フラッドベースの攻撃用にそれぞれ A, P, F と**図 3** のようにハッシュ値の定義付けを行った. この定義付けを用いて AS 間トレースバックの実運用を行った場合，以下のような役割が考えられる．

- Hash_F：特定の宛先アドレスに対してトラフィックを送信した全 AS を追跡する．
- Hash_P：特定の送信元アドレスから特定の宛先アドレス宛てに送られる，あるプロトコルに該当するフローの送信元 AS を追跡する．
- Hash_A：特定の宛先アドレス宛てに，ある TCP フローを転送した AS を追跡する．

本論文では，このハッシュ値定義を用いてサンプリング方式のトレースバックを行う．

3.4 フローダイジェスト方式を用いた AS 間相互接続アーキテクチャ

櫨山ら[6] は，AS マップを持たず eBGP の AS の隣接関係のみを元にして再起的に問い

Vol.2012-DPS-150 No.17
Vol.2012-CSEC-56 No.17
2012/2/29

表 2 実験機材

OS	Debian 4.0
CPU	Intel 6-Core Xeon X5670 x2
メモリ	48GB
HDD	SATA 500GB x2
Network	4 GbE for experiments
ルーティングソフト	Quagga 0.99.17-2, squeeze3

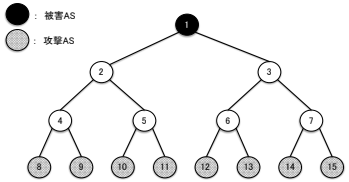

図 4 ツリー構造トポロジー

合わせを行う InterTrack を開発した．彼らはシミュレーションや実機を用いた実験, 評価を行っており，相互接続アーキテクチャの有効性を示している．AS 間トレースバックを目的としていること，また実装のソースコードが公開されており開発において応用しやすいことからフローダイジェスト方式のハッシュライブラリを InterTrack 上に実装した．

IP トレースバックが行われるにあたり，攻撃検知システムは TC(Traceback Client) に問い合わせを行うべきパケットのダイジェストを作成するよう依頼する．TC の作成するフローダイジェストは，DP を経由し，ITM(Inter-domai Traceback Manager) へ追跡メッセージと共に送られる．問題のパケットの流入 AS を BTM(Border Traceback Manager) に問い合わせ，かつ発信源が自分か否かを DTM(Domain Traceback Manager) に問い合わせる．BTM や DTM は具体的には ITM と追跡メッセージの送受信する役割を担い，問題のパケットの具体的な状況については BTS や DTS 等のシステムと連携するモデルとなっている．攻撃が流入してきたことが分かった場合，ITM は隣接 ITM に対して追跡メッセージを送信し，隣接 ITM でも同様の追跡作業を再帰的に行う．

この InterTrack でハッシュ値の生成するのは攻撃を検知して最初に追跡要求を送信する TC と，自 AS 内の監視を行っている BTM である．BTM は監視しているパケットのハッシュ値を生成し保持している．TC は攻撃パケットが発生するとパケットダイジェストの定義から攻撃パケットのハッシュ値を生成し，追跡要求と一緒にそのハッシュ値を BTM へ送信する．TC と BTM で共通のハッシュ関数，パケット情報を用いてハッシュ値を生成しているため，もし BTM が攻撃パケットのハッシュ値を保持している場合，TC が送って来たハッシュ値と一致する設計となっている．本研究ではハッシュ値生成のアルゴリズムは InterTrack と同じものを使用し，TC と BTM の共通のハッシュ値生成部分において，前説で定義したフローダイジェストの定義付けによってハッシュ値を生成する．

4. 実験結果

本実験は独立行政法人 情報通信研究機構 (NICT) 北陸リサーチセンター内のネットワークシミュレータ研究設備 (StarBED) にて行った．StarBED における実験機材の仕様を**表 2** に示す．疑似攻撃として ICMP Flood Ping 及び，netperf による TCP flood，UDP Flood を行った．**図 4** における被害者ノード AS1 に対して 3 ホップの距離にある 8AS から一斉に 1800 秒間攻撃し，10 回データをとった平均の発見 AS 数を計測した．

- 追跡モード：本論文では AS1 からの追跡要求を全ての隣接 AS に向けて送信する Flooding モードで実験を行った．
- サンプリングレート：R
  RFC3176[7] に示された, サンプリングレート R

$$R = \frac{TotalPackets}{TotalSamples} \tag{1}$$

  となるようにパケットをサンプリングするオプションが interTrack に実装されている．
- バルク数：b
  TC の追跡要求メッセージには，単一パケットの追跡だけではなく複数のパケットダイジェストを格納してフローとして追跡要求が行えるバルクモードが備わっている．実運用を考慮し，本論文では b の値を 1–60 までに制限した．

本実験ではバルク数 b を 60 に固定し，サンプリングレート R を変更した場合について実験を行った．R は設定値の下限である 1 から 100 ずつ上昇させていき，それぞれのハッシュ値生成手法で発見 AS が 0 になった値を上限として実験を行った． **図 5**，**図 6**，**図 7** はそれぞれ Ping flood 攻撃，UDP flood 攻撃，TCP flood 攻撃を行った場合の R を変化させた時の，SPIE 方式と hash_F，hash_P，hash_A の発見 AS 数を比較したグラフで，縦軸は平均発見 AS 数，横軸は R を示す．SPIE 方式は R を上げていくと徐々に発見 AS 数は減少し，$R = 1400$ で発見 AS 数が 0 となっているが，フローダイジェスト方式は R を増加させても発見 AS 数は減少することなく高い値を保持している．計測を続けた結果 R=20000 まで追跡成功率が減少することなく高い発見 AS 数を保持した．

次に R を 95 に固定し, バルク数 b を 10 から 60 の間で 10 ずつ値を増やしていき実験を行った. **図 8**，**図 9**，**図 10** はそれぞれ Ping flood 攻撃，UDP flood 攻撃，TCP flood 攻撃を行った場合のバルク数を変化させた時の，SPIE 方式と hash_F，hash_P，hash_A の発見 AS 数を比較したグラフで，縦軸は平均発見 AS 数，横軸はバルク数を示す．SPIE 方式はバルク数の低い時に発見 AS 数が下がっているが，フローダイジェスト方式ではバルク数に依存せず追跡が行えていることがわかる．

SPIE 方式では 1 つのパケットに 1 つハッシュ値を生成するため，R を上げた場合やバルク数を減らした場合，追跡を行うためのパケット情報が減少し，途中で追跡不可能となる．しかし，フローダイジェスト方式では 1 つの攻撃に対して 1 つのハッシュ値を生成するため，追跡を行うためのパケット情報がある程度減少しても，同じ攻撃の追跡に対してはパケット情報の数の影響を受けない．

Vol.2012-DPS-150 No.17
Vol.2012-CSEC-56 No.17
2012/2/29

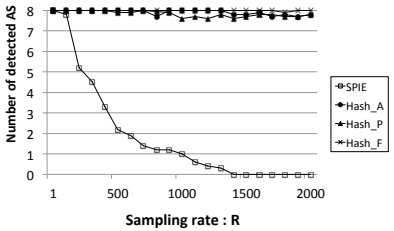

図 5 Ping flood 攻撃を行った際のサンプリングレートと発見 AS の関係

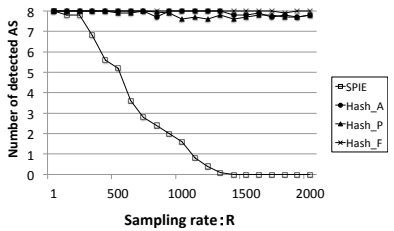

図 6 UDP flood 攻撃を行った際のサンプリングレートと発見 AS の関係

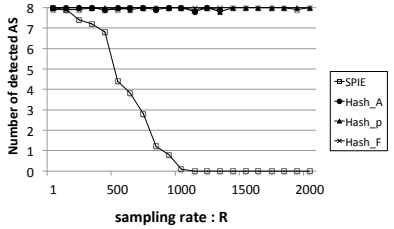

図 7 TCP flood 攻撃を行った際のサンプリングレートと発見 AS の関係

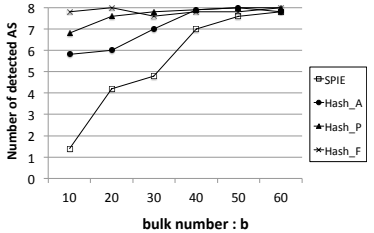

図 8 Ping Flood 攻撃を行った際のバルク数と発見 AS の関係

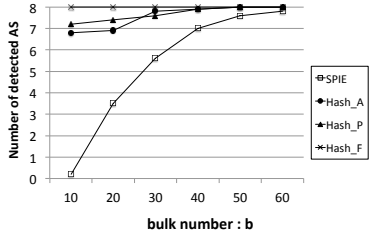

図 9 UDP Flood 攻撃を行った際のバルク数と発見 AS の関係

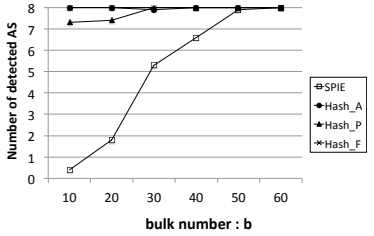

図 10 TCP Flood 攻撃を行った際のバルク数と発見 AS の関係

5. おわりに

本論文では，ダイジェスト手法サンプリング方式において，同じ DoS 攻撃のパケットに対して共通するハッシュ値を生成し，DoS 攻撃をフローとして追跡するフロー追跡について着目し，フルキャプチャ，sFlow，NetFlow で取得可能な値を用いて実攻撃を想定したフローダイジェスト方式を提案した．想定する攻撃はフラッドベース攻撃，プロトコルベースの攻撃，アプリケーションベースの攻撃の 3 種類の攻撃で，それぞれの攻撃に対して一意なハッシュ値を生成するハッシュ値の定義付けを行った．更にそのフローダイジェスト方式を IP トレースバックアーキテクチャである InterTrack にモジュール実装し，追跡性の検証をツリー構造トポロジーを用いて行った．検証の結果，本方式ではサンプリングレートやバルク数の影響を受けずに高い追跡成功率を保つことを明らかにした．既存のサンプリング方式では追跡を行うために実験的な高い値でしか，サンプリングレートやバルク数を設定できなかったが，提案方式を用いることで，より実用的な設定での追跡が行えるようになったと言える．

ただし，本研究では提案方式の追跡性を検証するために，ツリートポロジーの中で検証した結果であり，今後実インターネットで追跡性の検証を行う必要がある．実インターネットにおいて本方式での追跡を行った場合，攻撃ではない通常の通信トラフィックを誤って攻撃フローと判定する可能性があるため，本方式を単独で使用して追跡することは難しいと考える．従って，既存の単一パケット追跡と本方式との組み合わせについても検証する必要がある．また，本研究では 3 つの攻撃を用いて検証を行ったが，いずれの検証も一種類の攻撃が発生した場合である．同時に複数の手法を用いて攻撃が行われた場合，フローの定義付けがその攻撃の数だけ増加するため，パケット情報がより必要となりサンプリングレートやバルク数などの影響を受ける可能性が考えられる．これらを踏まえ，誤検知率の検証，提案方式と既存の単一パケット追跡の両方を効率的に取り入れた追跡方式の考案，複数の手法による攻撃が発生した場合の検証を今後の課題とする．

参 考 文 献

1) 村越 優喜 : 異種トレースバック方式混在環境下における AS 間トレースバック方式の特性評価に関する研究, 奈良先端科学技術大学院大学修士論文, 2009
2) Toshifumi Kai, Akirahashiguchi, Hiroshige Nakatani : Proposal for and evaluation of imploved method of hash-Based IP traceback system, Computer Science and its Applications, (2009).
3) A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer : Hash-based IP traceback, In Proceedings of ACM SIGCOMM ’01, pp. 3–14, (2001)
4) ArborNetworks, technical report : The growing threat of application-layer DDoS attacks, pp. 2, 2010
5) G. Blanc, H. Hazeyama, and Y. Kadobayashi : Flow Direction Inferring Using BGP Information Encapsulated in sFlow Packets, In Proceedings of JWIS ’07, pp. 103–118, (2007)
6) H. Hazeyama, Y. Kadobayashi, D. Miyamoto, and M. Oe : An autonomous architecture for inter-domain traceback across the borders of network operation, In Proceedings of ISCC’06, pp. 378–385, (2006)
7) P.Phaal et.al. InMon Corporation’s : sFlow : A Method for Monitoring Traffic in Switched and Routed Networks, RFC 3176(Information), (2001)