# 第 XIX 部

# IP トレースバック・システムの研究開発

# 第 19 部
# IP トレースバック・システムの研究開発

## 第 1 章　本報告の概要

IP トレースバックとは、パケットの通過した経路を特定する技術で、さまざまな方式や実装が存在し、不正アクセスへの対策として実用化が進んでいる。一方、種々の IP トレースバック・システムを相互接続し、インターネット上で横断的な IP トレースバックを実現するための運用アーキテクチャが必要とされている。

IP トレースバック WG では、インターネットでの実用性を重点においた IP トレースバック・システム相互接続アーキテクチャの提案および実装を行っている。

提案手法は、AS ごとに AS 内の全 IP トレースバック・システムを管理し、隣接 AS 間で AS 内への IP トレースバック要求や応答を交換することで、インターネット上での IP トレースバックを実現するものである。そして、本研究は、提案に基づく実装開発と実証実験を行うことで、インターネットでの IP トレースバック運用方式の標準となることを目指す[1]。

## 第 2 章　はじめに

分散型サービス妨害攻撃（DDoS 攻撃：Distributed Denial of Service Attack）は、インターネットにおける脅威となっている。たとえば、2004 年 1 月には、MyDoom ウイルスに感染した数十万ノードからの分散型サービス妨害攻撃によって、米 SCO 社の WEB サイトが停止に追い込まれ、深刻な被害を及ぼした [57]。

分散型サービス妨害攻撃とは、インターネット上に分散して設置された攻撃ノードから、被害ノードに対して大量のパケットを送ることによって、ネットワーク帯域やホストのリソースを奪い WWW（World Wide Web）や DNS（Domain Name System）といった正規のサービス提供を妨害する攻撃手法である。近年は、システムの脆弱性を利用して感染を広げるワームに、分散型サービス妨害攻撃を行う攻撃ノードの機能が実装され、その被害は大きくなり、防御は難しくなってきている。

このような分散型サービス妨害攻撃を防ぐには、攻撃ノードの設置を防ぐなどの事前対策と、発生後に迅速に攻撃を収束させるための事後対策が重要となる。事前対策としては、システムの脆弱性対策や、利用者のセキュリティ意識の向上といったことが挙げられる。しかしながら、CERT などの各インシデント機関を通して脆弱性に関する情報提供が行われているにも関わらず、管理の不十分なシステムが多数存在することや、ベンダーが脆弱性の対応をとる間にウイルスなどの急速な感染が進む現状から、事前対策のみで分散型サービス妨害攻撃を根絶することは困難である。事後対策とは、発生した分散型サービス妨害攻撃の攻撃ノードを特定し、インターネットからの隔離や経路上でのフィルタリングによって攻撃パケットの転送を防ぐことである。本研究において着目する IP トレースバック技術はこの手法の 1 つである。

IP トレースバックは、送信元が偽装された攻撃パケット群の転送経路を特定することができる技術で、追跡に要するコストの削減と分散型サービス妨害攻撃に対する事後対策に要する時間を短縮することができる。

IP トレースバックでは、図 2.1 に示された攻撃ノードから発信された攻撃パケットの流れを「攻撃フロー（Attack Flow）」とし、その攻撃フローが通過した経路を「攻撃パス（Attack Path）」とする。IP トレースバックは、攻撃フローに対する攻撃パスを特定する技術と定義できる。

IP トレースバックには、多数の手法が研究開発されており、その一部は、製品化されているものもある。したがって、現状では、IP トレースバック技術

1　本報告書は、IEICE-SCIS2005 で発表された論文を再構成したものである。

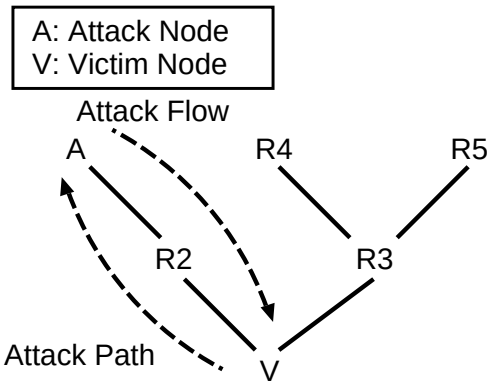


**図 2.1.**　IP トレースバック：攻撃パスの特定

自体は、十分に実用段階にあるものと言える。

しかしながら、インターネットは、国際的な通信インフラであり、商用 ISP や、企業、研究・学術機関といったさまざまなポリシをもった機関が相互に接続して成り立っている。また、現状提案されている IP トレースバック方式は、各方式ごとに導入コスト・運用コスト・追跡コスト・追跡精度などに長所短所が存在し、各ネットワーク環境に適した運用が行われると予想される。したがって、IP トレースバックの運用方法については、まだ検討段階であるといえる。

本報告書では、既存の IP トレースバックの運用アーキテクチャの概観を述べ、その問題点を指摘し、現在のインターネット上で、IP トレースバックの導入と相互接続運用が可能な手法を説明し、提案手法の実装アーキテクチャや各種プロトコルについて報告する。

以後、第 3 章では関連研究の説明、そして、第 4 章にて現在開発中の提案手法について述べ、第 5 章にてまとめと今後の課題を述べる。

## 第 3 章　関連研究

ここでは、既存の IP トレースバック技術を概観し、運用アーキテクチャの説明とその問題点を述べる。

### 3.1 IP トレースバック技術の現況

筆者らは、IP トレースバックに関する技術調査を行い文献 [347] においてまとめている。ここでは、提案されている種々の IP トレースバック技術の長所と短所を述べる。

既存の IP トレースバック技術は、その手法ごとに 4 つに分類することができる。

**1. リンク検査手法**

この手法は、既存のルータ上で交換されるトラフィックに対して、フィルタリングや経路制御を行い、対象となる攻撃フローの特定を行う。たとえば、ルータの持つフィルタリング機能を利用し攻撃フローを特定する手法 [290] や、ブラックホール経路を用いて攻撃フローを誘導することで特定する手法 [308] などがある。

この手法は、既存の機材を利用して追跡を行うため、その投資コストは最小限である。しかしながら、攻撃フローが発生している期間しかフローの追跡ができない点、管理者の行う手動追跡であるため追跡時間を必要とする点が問題である。

**2. 逆探知パケット手法**

ルータなどが攻撃パスを構成するために必要な情報を専用の IP パケット（これを、逆探知パケットという）に記録して送信し、これらから攻撃パスの構成を行う手法である [17, 341]。この手法は、既存バックボーン・システムに影響を与えることなく導入することが可能である。しかしながら、逆探知パケットによるインターネットのトラフィック増加と攻撃フロー特定に必要な時間のトレードオフが問題となる。

**3. マーキング手法**

各ルータが、IP ヘッダ中に IP トレースバックに必要な情報を記録することで攻撃パスを特定する手法である [273, 286]。IP トレースバックにともなう追加のトラフィックが一切発生しないが、攻撃ノードが偽造マーキングを生成した場合、攻撃パスの構築に必要な計算量が増大する点など抗トレースバック攻撃に対する防御が難しい点が問題である。

**4. ダイジェスト手法**

ハッシュ関数を利用した記憶容量の効率が高い手法で、ルータ上を通過するパケットを保存する。そして、パケットの保存記録から攻撃パスの特定を行う手法である [283]。この手法は 1 パケット単位での追跡が可能であるが、監視対象のインタフェースごとに、記録データを転送

するためのネットワーク帯域と保存のためのストレージ、そしてそれらの管理コストが必要となる点が問題である。

以上の既存研究は、いずれも IP トレースバックの有効性を示しており、インターネット上での利用が可能であるレベルにあるといえる。しかしながら、筆者らは、文献 [341] において、現実のインターネット上で IP トレースバックを運用するために解決しなければならない問題点を指摘している。

1. 現在のインターネットは、AS と呼ばれる経路制御の管理単位ごとに独立した管理・運用が行われている。このため、AS 間のトラブルへの対処は、AS 間での連携が必要となる。
2. IP トレースバック・システムの運用コストは、手法ごとに異なる。運用要件として、すべてのルータに対する改変を必要とする IP トレースバック手法や、IP トレースバック用の情報を保管するシステムの運用と保守を必要とする手法もある。したがって、ISP などの予算や運用規模に応じた IP トレースバック手法を運用する必要がある。
3. 日々攻撃手法が進化する状況を考えれば、攻撃者は、種々の IP トレースバック手法を分析しその弱点を狙った攻撃手法を開発すると予想される。したがって、それにあわせて、IP トレースバック手法も改良や変更を行わなければならない。

これらの問題から、単一の追跡手法による IP トレースバック・システムの運用ではなく、各ネットワーク上で展開される IP トレースバック・システムを相互接続するための運用アーキテクチャが必要である。

### 3.2 IP トレースバックの運用手法

ここでは、既存のインターネット上で IP トレースバック・システムを相互接続し、大域的な IP トレースバックを行う手法を述べ、その問題点を示す。

文献 [341] は、現在のインターネット上での経路制御機構に着目した IP トレースバック運用アーキテクチャの提案を行っている。この提案は、まず、攻撃フローの通過する AS を特定する AS トレースバック（図 3.1-(1)）によって、攻撃ノードの存在点を AS 単位で特定し、AS 内において、IP トレースバック（図 3.1-(2)）によって攻撃ノードの特定を行うという 2 段階にわけた手法である。このように、2 段階にわけて攻撃フローを特定することで、AS 特定から、フィルタリング実施などによる早急な被害の緩和、そして、攻撃ノード特定から被害の収束というように、段階的な事後対策が可能な手法となっている。

しかしながら、本手法は、インターネット全体での AS トレースバックの普及を原則としているため、その導入コストや、導入段階においては、限定的な攻撃フローの特定しかできない問題点がある。

一方、文献 [212] は、サービス妨害攻撃やワームなどの追跡と防御を行うために必要な情報を、IETF INCH WG がインシデント情報などの交換のために標準化している IODEF[191] フォーマット上で交換し、IP トレースバックを相互接続する RID（Real-Time Inter-Network Defense）手法を提案している。

本手法における攻撃パスの特定方法は、次の通りである。

被害ノードからの要求から、図 3.2 に示すように、被害ノードのネットワーク上で IP トレースバックを行い、攻撃フローの流入元ネットワーク境界点を特定する。次に流入元ネットワーク上で、同様に IP トレースバックによる攻撃フローの追跡を行う。この

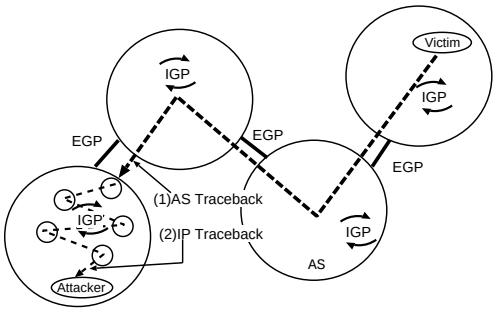


**図 3.1.** 階層型 IP トレースバック手法

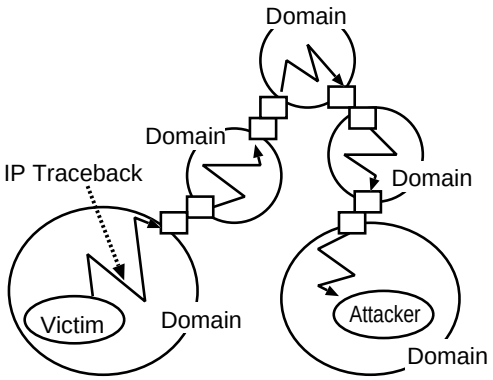


**図 3.2.** RID における IP トレースバック

再帰探索を攻撃ノード発見まで繰り返すことで、攻撃パスの特定を行う手法である。

この手法は、セキュリティインシデントの情報交換網上で、IP トレースバックを運用するため、汎用性に優れ、IODEF が導入されるならば、RID の導入コストは低いといえる。しかしながら、IODEF におけるインシデント交換のタイムスパンと、IP トレースバックにおけるタイムスパンは大きく異なっており、IP トレースバックでは、各トレースバック・システムの運用条件などから、リアルタイム性がより要求される。したがって、IODEF という汎用性のある運用枠組みでのリアルタイム性能へ及ぼす影響や、攻撃パスを全探索するモデルのため運用コスト（時間やリソース）が高くなり、実時間内に攻撃ノードを特定出来るかどうかという点についての考察が不十分である点が問題である。

### 3.3 現状の運用システムの問題点

現状の提案されている運用アーキテクチャは、導入に要する敷居が高く、また、インターネット全体での運用が成り立たなければ、その効果を発揮できない構成となっている。したがって、たとえば、IPv6 の普及のように、技術開発から導入までにかなりの時間を要することが予想される。

一方、インターネットにおける分散型サービス妨害攻撃の被害は深刻となっており、一刻も早い IP トレースバック・システムの運用が必要である。

このため、IP トレースバックをインターネット上で運用し、分散型サービス妨害攻撃に対する強力な対抗策とするためには、現状のインターネット運用に対して、導入負担の少ない運用アーキテクチャが必要であると言える。

## 第 4 章　提案手法

ここでは、筆者らが現在開発中の IP トレースバック運用アーキテクチャについて述べる。

### 4.1 提案手法の概要

本提案は、文献 [341, 342] において提案された階層型 IP トレースバックアーキテクチャを元に、より導入コストを低く、また、インターネットでの運用に則したアーキテクチャとなっている。

インターネットは、図 4.1 に示すように、IPv4・IPv6 アドレス空間ごとに、EGP（Exterior Gateway Protocol）による AS 間の相互接続、そして、AS 内の各ドメインが、IGP（Interior Gateway Protocol）によって相互接続することによって成り立っている。

そこで、本提案では、運用に対する影響を最小限とするために、運用構成を可能な限りシンプルな形として、低機能だが、インターネット上で展開しやすいことを念頭においた設計とした。

図 4.2 は、本提案手法における運用アーキテクチャを示したもので、インターネットの構成に則した形となっている。

本アーキテクチャでは、各ドメインで運用される IP トレースバック・システム（TS）を DTM（Domain

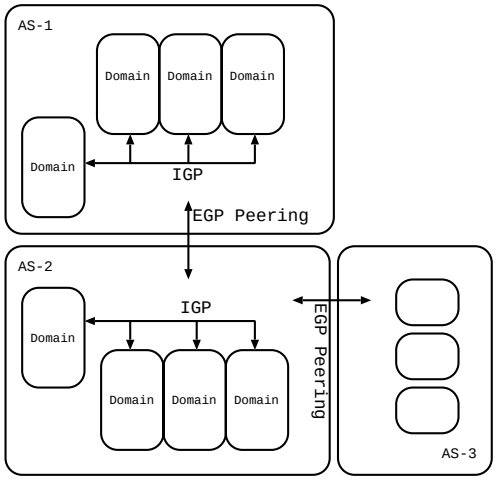


**図 4.1.**　インターネットの経路制御構成

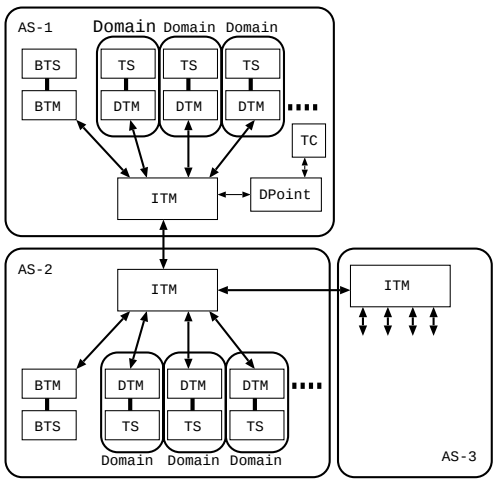


**図 4.2.**　提案手法のシステム関係図

Traceback Manager）によって、入出力を共通化し、AS 内の ITM（Internet Traceback Manager）が統括する。そして、ITM は、AS 境界におけるトレースバックを行う BTS（Border Traceback System）/ BTM（Border Traceback Manager）によって、攻撃フローの流入元・流出先 AS を特定する。これらのコンポーネントが、相互接続することによって、インターネット上で、横断的な IP トレースバックを行うことを実現する。

ここで、各コンポーネントの役割について述べる。

TS は、各ドメインの運用状況に合わせて運用される IP トレースバック・システムである。たとえば、ダイジェスト手法である Hash based IP トレースバックなどが、TS に該当する。TS は、DTM からの追跡要求や結果などのやり取りを行う。

DTM は、ITM からのトレースバック要求やその結果応答を TS の仕様に合わせてローカライズする。したがって、DTM は、TS ごとに異なる入出力仕様を共通化することにより、異なる TS が混在する運用環境を実現する。

ITM は、AS 内に 1 つ以上存在し、また、プライベートアドレス空間で運用されるネットワーク内にも存在する。ITM は、他 ITM や配下の DTM/BTS へのトレースバック要求や結果応答などのやり取りを行う。

BTS は、AS 境界においてトレースバックを行うことで攻撃フローを、1. AS 外への流出（図 4.3-(1)）、2. AS 外からの流入（図 4.3-(2)）、3. 流入に AS 内からの流出を含んだ流出（図 4.3-(3)）、4. トランジット（図 4.3-(4)）、5. 流入出なし、から判断する IP トレースバック・システムである。

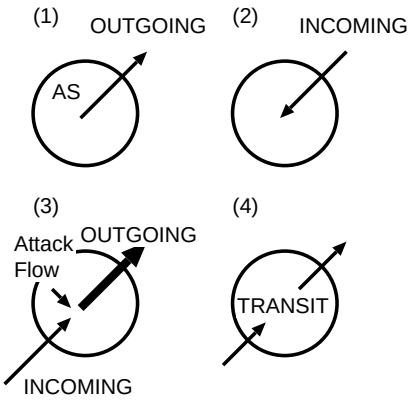


**図 4.3.** BTS の 4 つの機能

BTM は、DTM と同様に、BTS の入出力を共通入出力化し、ITM との入出力を実現するものである。

D Point（Decision Point）は、IP トレースバックの実行を要求する TC（Traceback Client）からの追跡要求を受け取り、IP トレースバック・システムの運用ルールや実行条件を満たすクライアントかどうかなどの条件を元に、追跡を開始するかどうかを決定するシステムである。IP トレースバックを実行する場合、TC から提供される攻撃フローの情報などを ITM に伝える。

また、本手法では、段階的導入において、ドメイン内に TS や、AS 内に BTS がない場合は、導入コストがきわめて低いスタブシステムが、追跡不能とする応答を行うことによって、探索結果の品質は下がるが、システム全体が機能する構成となっている。

### 4.2 各コンポーネントの動作

ここでは、本提案手法における IP トレースバックの流れについて述べる。TC からの攻撃フロー追跡要求からその結果を受け取るまでの流れは、次のようになっている。

1. 被害ノード（Victim）が IP トレースバックに必要な情報を提出（図 4.4-(1)）。
2. TC が D Point に対して攻撃フローのトレースバック要求を送信（図 4.4-(2)）。
3. D Point は、受付時のタイムスタンプを付加し、ポリシに基づき実行の可否を判断し、可の場合、ITM に対してトレースバック要求を行う（図 4.4-(3)）。
4. ITM は、D Point からトレースバック要求に対してシーケンス番号を発行し応答する。発行したシーケンス番号を D Point を介して TC に転

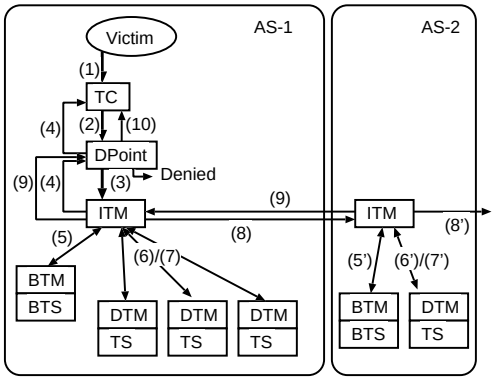


**図 4.4.** 各コンポーネントの動作シーケンス

送（図 4.4-(4)）する。

5. ITM から BTM へ、攻撃フローの流入・流出状況の問い合わせを実施（図 4.4-(5)）し、その結果にしたがって次の動作を決定する。
   (a) 流出の場合、または流入出なしかつ D Point からの要求の場合、攻撃フローは、自 ITM 配下に存在する。ITM は DTM へトレースバック要求（図 4.4-(6)）と応答（図 4.4-(7)）を受け取る。そして、要求元へ結果を応答する（図 4.4-(9)）。
   (b) 流入のみ、またはトランジットの場合、流入元 AS の ITM へ追跡要求を行う（図 4.4-(8)）（流入元 AS の ITM が (5) を実行）。結果を取得し、要求元へ結果を応答する（図 4.4-(9)）。
   (c) 流入出の場合、ITM は、DTM へトレースバック要求を行う。（図 4.4-(6)）。結果を取得し（図 4.4-(7)）、次に、流入元 AS の ITM へトレースバック要求を行う（図 4.4-(8)）（流入元 AS の ITM が (5) を実行）。結果を取得し、DTM での結果と合成し、要求元へ結果を応答（図 4.4-(9)）する。
6. D point に ITM からの結果が応答されたら、TC へ結果を応答する（図 4.4-(10)）。

以上に示すように、本提案では、TC からの追跡要求を元に、再帰的に探索を行い攻撃フローの特定を行うシステムとなっている。しかしながら、BTS によって、攻撃フローの流入出を判断することによって、RID に比べてより効率の高い IP トレースバック運用アルゴリズムを有している。

### 4.3 各コンポーネント間の通信方法

現在、RID においては、通信プロトコルとして、SOAP や BEEP を利用し、メッセージは、XML を用いる構成とすることが検討されている。

本提案では、通信プロトコルとして、BEEP による実装検討を行ったが BEEP のライブラリなどの利用コストが高いという結論に至った。本提案方式は、各ドメインや AS がネットワークの相互接続の運用関係に則して通信するアーキテクチャとなっているので、通信相手を特定することが可能である。したがって、通信プロトコルは、IPsec 上の TCP 通信を利用し、その上で交換されるメッセージは、XML で構成することとしている。

表 4.1 は、TC から D Point へトレースバック要求（図 4.4-(2)）を行うときに送信されるメッセージで、D Point は、ITM から受け取ったシーケンス番号（図 4.4-(4)）を表 4.2 に示すメッセージで応答するようになっている。

**表 4.1.**　トレースバック要求メッセージ例

```
<ClientTraceRequest>
  <SourceClientID>
    IP address
  </SourceClientID>
  <DestinationDecisionPointID>
    IP address
  </DestinationDecisionPointID>
  <PacketDump iftype="IF">
    HEX DATA
  </PacketDump>
  <Options>
  </Options>
</ClientTraceRequest>
```

**表 4.2.**　トレースバック応答メッセージ例

```
<ClientSequcenceReply>
  <DestinationClientID>
    IP address
  </DestinationClientID>
  <SourceDecisionPointID>
    IP address
  </SourceDecisionPointID>
  <Origin>Origin Issuer ITM ID</Origin>
  <SequenceNumber>
    sequence number
     issued by Origin ITM
  </SequenceNumber>
  <Options>
  </Options>
</ClientSequcenceReply>
```

## 第 5 章　おわりに

本報告書では、現在の IP トレースバック研究に関する現況や運用システムについて述べ、IP トレースバック技術は、実用段階にあるが、インターネットで運用するためのアーキテクチャが不十分であることを指摘した。そして、インターネットの運用形態に則した、非常にシンプルで導入コストが低い IP トレースバック運用アーキテクチャを提案し、その

概要を述べた。

本提案手法は、現時点では、机上検証段階であり、その有効性については、実証に至っていない。したがって、筆者らは、提案手法の有効性を示す方法として、実装を利用した検証を考えている。

本研究の目的は、現在のインターネットにおいて、即効性をもったIPトレースバック・システムの運用可能なシステムの開発である。したがって、実証実験を通した検証が重要であるといえる。

筆者らは、現在、XMLメッセージの策定と、ITM/BTS/DTSなどのプロトタイプ開発を進めており、2005年1月中に本アーキテクチャの実装を完了する予定である。

そして、2005年2月下旬に、NICT北陸IT研究開発支援センターにあるインターネットエミュレーション環境[220]上で、仮想インターネット環境を構築し、実装を用いた検証実験を行う予定である。

実験によって有効性を示した後に、実インターネット上での展開や、より効率のよい探索といった洗練されたアーキテクチャを目指して改良を進めていく予定である。