修士論文　　2005年度 (平成17年度)

ベイズ推測を用いた不正パケット スコアリング手法の研究

慶應義塾大学大学院 政策・メディア研究科

白畑 真

修士論文要旨 2005 年度 (平成 17 年度)

ベイズ推測を用いた不正パケットスコアリング手法の研究

論文要旨

ワームや DoS 攻撃等の不正トラフィックを識別するための手法としては，IDS 等が広く知られている．しかし，既存の手法では未知の攻撃手法を検知できない，広帯域ネットワークへの対応が困難であるといった課題が存在する．

本研究では，これらの問題を解決するため，パケットのヘッダフィールドの値のみを用いて未知の攻撃を含む攻撃を検知し，広帯域ネットワークに適用可能な不正パケット識別手法を提案する．

本論文ではハニーポットで収集したトラフィックと，研究・教育用ネットワークのトラフィックに対して分析を行い，それぞれのトラフィックのヘッダフィールドの値に異なった傾向が存在する点に着目し，ベイズ推測によりスコアを算出する手法を提案した．

そして，提案手法を用いて研究・教育用ネットワークのトラフィックのスコアリングを行い，定量評価および定性評価を通じて提案手法を評価した．

キーワード
1. インターネット，2. セキュリティ，3. パケットスコアリング，4. 侵入検知

慶應義塾大学 政策・メディア研究科

白畑 真

Abstract of Master's Thesis

Academic Year 2005

Research of Malicious Packet Scoring Method based on Bayesian Analysis

IDS is a well known method for identifying malicious traffic such as Worms and DoS attacks. However, IDS is difficult to be deployed in high speed networks, and still cannot detect unknown attack methods.

In the research, we proposed a new method to solve the issues mentioned above based on the values of IP packets' header fields only.

From deep analysis between the traffic collected with Honeypot and the traffic from real research & education network, we discovered malicious packets have distinctive patterns in the header field. Using this pattern with Bayesian theory, we proposed a scoring method for IP packets.

For evaluation, we scored the packet of the research network using our proposed method and evaluated in both qualitative and quantitative way. Based on these evaluations, we found that this method has a advantage in computational resources, compared to existing IDS methods.

Keywords :
1. Internet, 2. Security, 3. Packet Classification, 4. Intrusion Detection

Keio University, Graduate School of Media and Governance

Shin SHIRAHATA

目 次

図 目 次

表 目 次

第1章　序論

1.1　研究の背景

近年，インターネットはその利用範囲の拡大に伴い，社会において重要なインフラストラクチャとなりつつある．このため，以前と比較してインターネットの可用性に対する要求が高まっている．

インターネットは，エンドノードとエンドノードの通信を中継する中継ノード，そしてノード間を接続するネットワークによって構成されている．通常，単一のネットワークには多くのホストが接続されているため，中継ノードやネットワーク自体の可用性は，多くのエンドノードの接続性に影響を及ぼす．

例えば，2003年1月に大流行したSQL Slammerワームは，攻撃のためのパケットを大量に送信したため，一部のネットワークの輻輳やルータへの過負荷を引き起こした．この結果，通信品質の低下やネットワークの停止などの影響が発生した [1]．

インターネットが社会インフラとして利用されるようになるにつれ，ネットワークの可用性がますます重要になっている．2000年2月に発生したDDoS(分散サービス妨害) 攻撃では，Amazon.comやYahoo!，eBayなどのWebサイトに長時間アクセスできなくなり，電子商取引に大きな影響を及ぼした [2]．DDoS攻撃は，攻撃者が標的に対して直接攻撃を行うのではなく，攻撃者の制御下にある多数のホストを遠隔制御し，これらのホストが間接的にDoS攻撃を行うため，大規模な被害が発生する．

今後，大量の不正トラフィックが分散的に発生する事態が生じた場合，これまでにない甚大な被害の発生が予想される．インターネットを構成するネットワークにおいては，ADSL，FTTHなどのブロードバンドの普及により，アクセスネットワークの急速な広帯域化が進んでいる．ADSLでは最大50Mbps，FTTHでは100Mbps，一部では1Gbpsもの帯域幅のサービスが提供されている．一方，ISP(インターネットサービスプロバイダ) のバックボーンネットワークやトランジットにおいては，100Mbpsから1Gbps，最大でも10Gbps程度となっており，アクセスネットワークの急速な広帯域化に見合うだけの回線帯域をバックボーンネットワークにおいて確保できていないのが現状である．

現時点において多くのISPでは，統計多重効果によりアクセスネットワークの帯域が同時に消費されない前提でバックボーンネットワークが設計されている．この設計においては，平常時においては何ら問題は生じないが，ウィルスやトロイの木馬等によって形成されるボットネット等により，自律分散的なDDoS攻撃が実施された場合や，ワーム等が急激に拡散した場合には，設計時に想定していなかったほどのトラフィックが多数のアクセス回線に対して同時多発的に流入し，バックボーンネットワーク，ひい

てはインターネット全体の可用性に影響を及ぼす危険が想定される．

従って，未知の攻撃手法による不正トラフィックがネットワークインフラストラクチャに与える脅威に対抗する防御手法の開発が必要である．

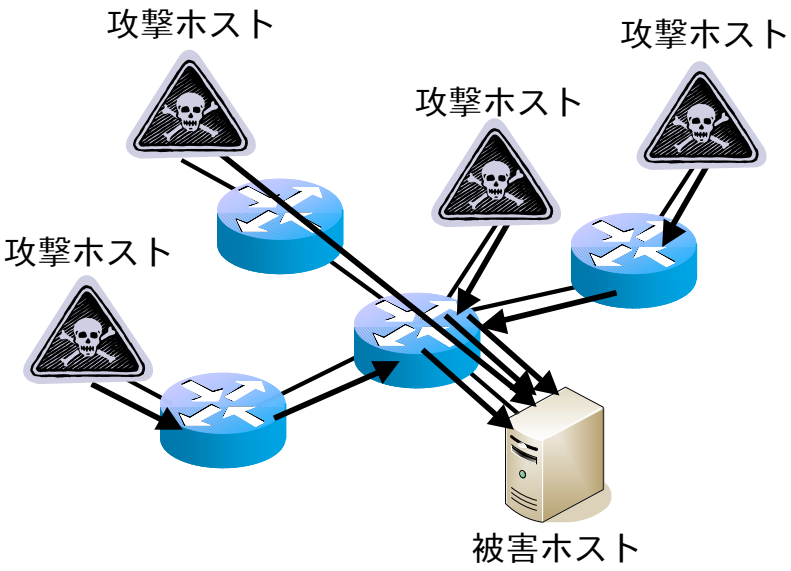

図 1.1: DDoS 攻撃のしくみ

1.2 本研究の目的

本研究では，トラフィック特性に基づき，パケットの潜在的なリスクをパケットに対するスコアリングとして表現する手法を提案する．本手法により，ワームや DoS (サービス妨害) 攻撃，DDoS 攻撃などの不正トラフィックをスコアリングを行い，トラフィックを制御することで，ネットワークインフラストラクチャを保護することが期待される．

既存のパケット識別手法では，IDS/IPS はトラフィックの正規化を行うため，広帯域ネットワークへの対応や，多くのホストを監視対象とした場合のスケーラビリティに問題がある．

また，セキュリティに関するトラフィックの識別と制御は二元論的なアプローチに基づいて行われてきた．例えば，不正検知方式を用いた IDS/IPS では，シグネチャに基づきトラフィックが侵入であるか否かを識別している．Firewall もアクセス制御リスト (ACL) に基づき通過させるべきトラフィックか否かを識別している．

しかし，二元論的なアプローチでは，ネットワーク管理者はトラフィックを制御する手法としてパケットを転送するか，あるいは破棄するかの二択を迫られる．従って，シグネチャの存在しない未知の攻撃手法，あるいは ACL で許可したトラフィックが不正トラフィックであった場合の悪影響が懸念される．

このため，本手法では未知の攻撃を含む不正トラフィックを識別するために，ベイズ推測を用いる．本手法では，あらかじめ通常ネットワークとハニーポットの両方からトラフィックを収集し，そのトラフィックのヘッダ情報を事前確率のパラメータとする．そして，トラフィックのヘッダ情報をパラメータとしたベイズ推測を用いて事後確率を求めることにより，パケットに対してスコアリングを行う．

これにより，ネットワーク管理者にこれまでの識別手法とは異なる判断基準を提示でき，リスクに応じた動的な優先制御，TCP ウィンドウサイズ制御，通信可能ノード数制御，遅延制御，帯域制御などのトラフィック制御手法への応用が期待できる．

本研究では，他のネットワークに対して対外接続性を提供するバックボーンネットワークを想定環境とする．バックボーンネットワークは，多くのトラフィックを転送するため，下流のネットワークの帯域に比べて広帯域 (1Gbps〜10Gbps) であり，単に広帯域であるだけでなく，バックボーンを介して多くのホストが通信するため，多くのトラフィックフローを転送している状況を想定する．

1.3 本論文の構成

本論文は第 2 章において，既存のパケットスコアリング・制御機構の位置づけを整理し，Firewall および IDS/IPS をパケットスコアリング機構としてとらえる．そして，これらの機構がバックボーンネットワークにおいて，不正トラフィックを抑制するためのパケットスコアリング機構として不十分である点を述べる．

第 3 章においては，関連研究の紹介と，それらが抱える問題点を整理する．

第 4 章では，問題点の解決・今後の要求を満たす手法の設計を述べ，第 5 章では，本研究で用いたハニーポット技術の詳細について述べる．

第 6 章において，研究・教育ネットワークで収集されたトラフィックおよびハニーポットで収集されたトラフィックの分析結果について取り上げ，相違点を分析する．

第 7 章では提案手法のプロトタイプ実装に関して述べる．

さらに，第 8 章において，提案した手法が要件を満たし，問題を解決しているかを軸に評価する．

最後に第 9 章で本研究の成果と今後の課題をまとめる．

第2章　パケット識別機構の概要と課題

2.1　パケット識別機構

本章ではまず，既存のパケット識別手法と，これらを実装した機構について整理する．次に，これらの機構がネットワーク境界において，不正トラフィックを抑制するために不十分である点を述べる．最後に，問題の解決に必要な条件を定める．

2.1.1　パケット識別の定義

本研究では，パケット識別を事前に定義されたパケットのヘッダもしくはペイロードの内容，あるいはそれらの組み合わせに条件により，パケットを区別する仕組みであると定義する．

2.1.2　不正トラフィックの定義

本研究では，ユーザに供しない目的のサービスにおいて，外部から当該サービスに対するアクセスに伴うトラフィック，あるいは通常のサービス利用に対して過剰なトラフィックを不正トラフィックと定義する．ここでは，ワームの活動に伴うトラフィック，(D)DoS のアタックトラフィックを不正トラフィックとして想定する．

2.1.3　パケット識別の対応の必要性

パケット識別は，ネットワーク運用のさまざまな局面で必要となる技術である．IP ネットワークにおいてもっとも基本的なパケット識別処理は，ルータにおけるパケット転送プロセスである．パケット転送では，ルータが宛先 IP アドレスを参照し，経路表に基づいてパケットの出力先インタフェースを決定する．

また，異なったアプリケーションに対して異なったサービスを提供する Qos (Quality of Service) は，遅延やパケットロスの影響を受けやすい VoIP やビデオ会議などのアプリケーションの通信品質を確保するために重要である．QoS では，サービスを提供するトラフィックを特定するために，パケット識別が必須である．

さらに，ネットワークの境界においてトラフィックを制御する，いわゆる境界型セキュリティ機構も，パケット識別機構の一種である．Firewall は管理者の設定したアクセス制御リストに従ってパケットを識別し，不正検知型の IDS/IPS は攻撃トラフィッ

表 2.1: 宛先 IP アドレスに基づくパケット識別

参照フィールド	ヘッダ: 宛先 IP アドレス
データの正規化	不要
フロー状態の保持	不要
必要な計算機資源	きわめて低い
不正トラフィックの識別	ハニーポットを除き困難

クのパターンと管理者の設定したポリシに従ってパケットを識別する．すなわち，境界型セキュリティ機構では，破棄すべきトラフィックを判断する手段としてパケットの識別を行っているといえる．

また，ネットワーク監視においては，プロトコルの種類やトラフィック量，あるいはトラフィックの種類を分類するためにトラフィック識別が必要となる．

このように，パケット識別は QoS ばかりではなく，ルータや Firewall，ネットワーク監視など，ネットワーク運用に関わるさまざまな場面で運用されている．

2.1.4 宛先 IP アドレスに基づくパケット識別

表 2.1 に宛先 IP アドレスに基づくパケット識別の概略を示す．宛先 IP アドレスに基づく識別は，IP ヘッダの特定のフィールドを参照するだけ実現できるため，低い計算機資源のコストでトラフィックを識別できる．しかし，宛先 IP アドレスは通信先ノードを特定するだけの識別子であるため，ある通信先ノード宛のトラフィックが全て不正トラフィックである場合を除き，不正トラフィックの識別には利用できない．従って，宛先 IP アドレスがハニーポットである場合を除き，宛先 IP アドレスを元にトラフィックの識別はできない．

2.1.5 ネットワーク層およびトランスポート層ヘッダに基づくパケット識別

トランスポート層ヘッダに基づくパケット識別手法として，QoS 技術である IntServ[3] および DiffServ[4]，ネットワーク型 Firewall を挙げる．

2.1.6 QoS (Quality of Service)

インターネットの普及と利用されるアプリケーションの多様化に伴い，ビデオ会議や音声通信などのアプリケーションが必要とする QoS を提供する必要性が高まってきた．このため，インターネットにおいて多様なサービス品質，すなわち QoS を実現するアーキテクチャが登場した．

従来のインターネットは，ベストエフォートのネットワークであり，エンドノード間の IP 到達性以上のサービスを提供しない．このため，インターネットにおいてはパケットフローの平均レート，ピークレート，パケットの遅延時間，ジッタ，パケットロ

表 2.2: IntServ におけるパケット識別

参照フィールド	送信元 IP アドレス，宛先 IP アドレス， トランスポート層プロトコル，送信元 TCP/UDP ポート番号， 宛先 TCP/UDP ポート番号
データの正規化	不要
フロー状態の保持	必要
必要な計算機資源	中程度
不正トラフィックの識別	IP ヘッダを対象とする．定義付けに依存

スなどが全て QoS の要素となる．そして，このようなフローに対して指定された品質を維持すると同時に，経済的に転送する技術を QoS 技術という．

本研究では，ワームや (D)DoS などの不正トラフィックをアプリケーションとして識別することを目的とする．本研究により識別された不正トラフィックと，非不正トラフィックに対して異なった QoS を設定することにより，ワーム等のトラフィックによるネットワークの輻輳を回避，あるいは軽減する応用が可能である．また，意図的に低いピークレートや，高いパケットロス率を設定することにより，ワームの感染速度を低下させたり，(D)DoS の影響を緩和させる応用も考えられる．

2.1.7 IntServ (Integrated Services)

IntServ は，シグナリングプロトコルである RSVP (Resource reSerVation Protocol)[5] によりネットワーク中の資源を予約することにより，QoS を実現する技術である．IntServ では，各フローに対して QoS を提供することを目指しており，経路上の機器は全てのフローの状態を保持し，パケットを転送するたびにパケットのヘッダの内容とフローの状態を確認する必要がある．

従って，IntServ をバックボーンネットワークのような多くのホストのトラフィックを通過させるネットワーク上に適用しようとした場合，管理しなければならないフローの数も膨大になり，中継ノードの処理能力を必要とする．また，RSVP は，フローの状態を保持するために 30 秒ごとにメッセージを送出するため，オーバーヘッドが大きい．

このため，IntServ/RSVP は，中継ノードがフローの状態を管理するため，スケーラビリティがない，シグナリングメカニズムが複雑すぎる，サービスプロバイダのビジネスモデルとの整合性に欠けるといった理由により広く普及することはなかった．

表 2.2 に IntServ におけるパケット識別の概略を示す．IntServ はフローに基づいた QoS の実現する機構であり，フローの状態を保持するにはヘッダフィールドの値を解釈する必要がある．従って，不正トラフィックのネットワーク層およびトランスポート層のヘッダ情報を定義することにより，IntServ を用いて不正トラフィックを識別できると考えられる．

ただし，ネットワーク層およびトランスポート層ヘッダ情報だけで不正トラフィックを識別する手法には限界がある．次節において本問題の詳細を述べる．

表 2.3: Multi Field Classifier におけるパケット識別

参照フィールド	送信元 IP アドレス，宛先 IP アドレス， 送信元 TCP/UDP ポート番号，宛先 TCP/UDP ポート番号， 受信インタフェース，パケット長，IP Precedence， DSCP，CoS，MPLS ラベルなど
データの正規化	不要
フロー状態の保持	不要
必要な計算機資源	低い
不正トラフィックの識別	IP ヘッダを対象とする．定義付けに依存

2.1.8 DiffServ (Differentiated Service)

IntServ/RSVP の欠点を克服し，より現実的な QoS 機構として DiffServ が提案された．DiffServ では，エンドノード間のサービスを規定するのではなく，QoS を実現するフレームワークとコンポーネントの一部を定義している．

DiffServ においては，DS ドメインとよばれる閉じたネットワーク内にパケットが到着した際にパケットを識別し，クラス分けを行い，IPv4 の場合には ToS フィールド，IPv6 の場合には Traffic Class フィールドに対して DSCP (DS Code Point) を書き込む．

ある QoS 処理が必要なフローに対しては，同一の DSCP を付与することにより，フローの集合体である同一の DSCP を持つトラフィックを対象に QoS を提供する．中継ノードは，この DSCP により動作を決定するため，シグナリングが不要になるとともに，個々の中継ノードがフローの状態を確認する必要はなくなり，スケーラビリティが拡大する．

DiffServ のアーキテクチャにおいては，Classifier がパケットを識別し，クラス分けを行う．表 2.3 に Multi Field Classifier におけるパケット識別の概略を示す．Classifier には，コアノードで利用される “BA (Behavior Aggregate) Classifier” と，エッジノードで利用される “Classifier” の二種類があり，前者は DSCP の値のみを参照するのに対して，後者は送信元 IP アドレス，宛先 IP アドレス，送信元 TCP/UDP ポート番号，宛先 TCP/UDP ポート番号，受信インタフェース，パケット長，IP Precedence などのパケットヘッダを参照する．

DiffServ は IntServ とは異なり，フローの集合体に対してクラス分けを行う．例えば，送信元ポート番号が 80 番，宛先ポート番号が 1024 番以上というフローの集合に対してクラス分けを行うことができる．

不正トラフィックの識別のために，Multi Field Classifier を利用するには，不正トラフィックの特性をあらかじめ定義する必要がある．Multi Field Classifier では，IP アドレス，ポート番号といったフロー情報に加え，パケット長や IP Precedence などのヘッダ情報や受信インタフェース情報などを用いることができるため，より詳細な条件を定義できる．ただし，基本的にパケットのヘッダ情報にとどまるため，パケットのペイロードのようなアプリケーション層の情報に基づいた不正トラフィックの識別はできない．

表 2.4: ステートレス型 Firewall パケット識別

参照フィールド	送信元 IP アドレス，宛先 IP アドレス， トランスポート層プロトコル，送信元 TCP/UDP ポート番号， 宛先 TCP/UDP ポート番号
データの正規化	ネットワーク層データの正規化が必要
フロー状態の保持	不要
必要な計算機資源	低い
不正トラフィックの識別	IP ヘッダを対象とする．アクセス制御リストに依存

表 2.5: ステートフル型 Firewall パケット識別

参照フィールド	送信元 IP アドレス，宛先 IP アドレス トランスポート層プロトコル，送信元 TCP/UDP ポート番号 宛先 TCP/UDP ポート番号
データの正規化	ネットワーク層およびトランスポート層データの正規化が必要
フロー状態の保持	必要
必要な計算機資源	やや低い
不正トラフィックの識別	IP ヘッダを対象とする．アクセス制御リストに依存

2.1.9 Firewall

Firewall はあらかじめ定義されたポリシに従い，トラフィックの通過の可否を判断する機構である．パケットフィルタ型 Firewall では，DiffServ 同様のパラメータを用いてトラフィックの識別を行っている．具体的には，送信元 IP アドレス，宛先 IP アドレス，送信元 TCP/UDP ポート番号，宛先 TCP/UDP ポート番号に基づきに通信の可否を判断している．

DiffServ がフローの集合に対してクラス分けを行うのと同様，Firewall もアクセス制御リストに従い，パケットの通過，もしくは破棄を決定する．100%のパケットロスを QoS の一手法とするならば，Firewall も広義の QoS 技術であるといえる．

Firewall がトラフィックを識別する対象は，Firewall の実装によってことなる．フローのステートを持たないステートレス Firewall は，個々のパケットに対してトラフィックを識別する．ステートレス型 Firewall におけるパケット識別の概略を表 2.4 に示す．

一方，ステートフルインスペクションを行うネットワーク型 Firewall は TCP などトランスポートのフロー状態を保持し，状態に基づいてトラフィックを識別する．ステートフル型 Firewall におけるパケット識別の概略を表 2.5 に示す．

ステートレス型 Firewall のアクセス制御は，外部から TCP の送信元ポート 80 番であるパケットを常に通過させるといった設定に限定される．従って，送信元ポート番号を意図的に 80 番に設定した接続であれば，ネットワークの外部から内部に対する接続であっても，接続を確立させることができる．

一方，ステートフルインスペクション型 Firewall では，TCP セッションといった通信の方向とフローの状態を保持するため，内部から外部に対して確立された接続と，それに対する応答以外のパケットを遮断する．従って，ステートフルインスペクション型 Firewall では，アクセス制御リストを迂回するよう意図的にポート番号を設定した外部からの通信を破棄できる．ただし，セッション情報を保持しなければならないた

め，Firewall はフローの状態を保持し，パケットを転送するたびにパケットのヘッダの内容とフローの状態を確認する必要がある．

いずれの場合でも，Firewall では，アクセス制御リストで許可されたトラフィックが正当なサーバへのアクセスに伴うトラフィックなのか，サーバの脆弱性を狙った攻撃に伴うトラフィックかを区別できない．図 2.1 に，Web サーバのポート 80 番宛のトラフィックを許可するポリシを設定した Firewall が，攻撃の通過を許す例を示す．

このように，Firewall を初めとするネットワーク層およびトランスポート層ヘッダに基づくパケット識別手法では，トランスポート層以下のヘッダ情報によって判断を行っているため，アプリケーション層における攻撃を許すなど，ネットワーク管理者の視点からすれば好ましくない判断が行われる問題がある．

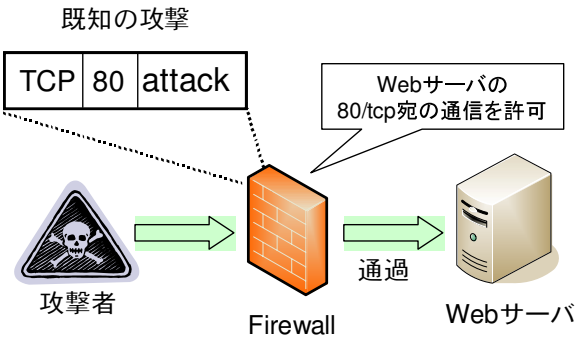

図 2.1: Firewall を通過する攻撃

2.2　既存手法の問題点

これまで，正当なアプリケーションの利用のような適格トラフィックと，ワームや DoS などの不適格トラフィックを識別し，制御する手法として，Firewall や IPS (Intrusion Prevention System) が用いられてきた．本節では，これらのトラフィック識別・制御技術とその問題点について述べる．

2.2.1　アプリケーション層のデータグラムに基づくパケット識別

ネットワーク層およびトランスポート層ヘッダに加え，アプリケーション層のデータグラムに基づいてパケット識別を行う手法について述べる．

2.2.2　IPS

IPS は，主に不正検知方式 (Misuse Detection) の IDS(Intrusion Detection System) をベースに開発されている．多くの IPS/IDS はネットワーク層，トランスポート層，ア

プリケーション層の各レイヤにおいてトラフィックの正規化処理を行っている．

ネットワーク層での具体例としては，フラグメント化されたパケットの再構築，リオーダリングされたIPパケットの並び替えが挙げられる．トランスポート層では，TCPストリームの再構成，タイムアウト処理が必要とされる．さらに，アプリケーション層では，プロトコルやサーバアプリケーションごとに異なる正規化を行う必要がある．

例えば，

```
/cgi-bin/phf.cgi
```

へのアクセスを不正と検知するシグネチャでは，単に

```
http://www.example.com/cgi-bin/phf.cgi
```

というURIへのアクセスを検知するだけでは不十分である．多くのWebサーバでは，

```
http://www.example.com/cgi-bin/dummy/../phf.cgi
```

のような，冗長なパス名を含むファイル名表記，あるいは

```
http://www.example.com/%2f%63%67%69%2d%62%69%6e%2f%70%68%66%2e
%63%67%69
```

のような文字コードをエンコードした表記であっても，/cgi-bin/phf.cgiというリソースへのリクエストとして扱う．従ってIDSは，Webサーバが解釈可能な全てのURI表記方法を，シグネチャの表記に正規化する必要がある．

既存のIDSやIPSは，このような複雑な正規化処理に多くの計算機資源を要求するため，広帯域なネットワーク環境への適用が難しい．また，データの正規化は，究極的にはエンドノードで稼働しているアプリケーションごとのエミュレーションを必要とするため，新たなプロトコルや実装が登場するごとに対応が求められる．表 2.6 にIDS/IPSにおけるパケット識別の概略を示す．

不正検知方式のIPS/IDSは，図 2.2 に示したように，シグネチャと呼ばれる攻撃のトラフィックパターンのデータベースをあらかじめ保持しておき，当該データベースに存在するか否かを元に，トラフィックを判断している．このため，既知の攻撃手法しか検知できない．また，IPS/IDSにおいて正規化が不十分であると，IDSの検知を迂回し，その機能を無効化できることが指摘されている [6]．

2.2.3 Network-Based Application Recognition (NBAR)

NBAR[7]は，TCP/UDPのポート番号だけではなく，HTTPのURL，ホスト，MIME (Multipurpose Internet Mail Extension) タイプなどを考慮したパケット識別機構である．NBARにおけるパケット識別の概略を表 2.7 に示す．

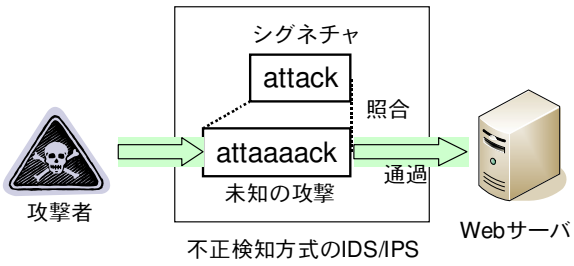

図 2.2: IPS の動作概略

表 2.6: IDS/IPS のパケット識別の概略

参照フィールド	送信元 IP アドレス，宛先 IP アドレス トランスポート層プロトコル，送信元 TCP/UDP ポート番号 宛先 TCP/UDP ポート番号，IPID，TTL，ペイロード全て
データの正規化	ネットワーク層，トランスポート層， アプリケーション層のデータの正規化が必要
フロー状態の保持	必要
必要な計算機資源	極めて高い
不正トラフィックの識別	不正トラフィックを定義したシグネチャおよび 管理者の定義したポリシーによる

NBAR を不正パケット識別のために利用できる場合は，特定のサービス，あるいは URL やホスト，MIME タイプが不正トラフィックであると決定づけられる場合に限定される．このような手法は，Microsoft SQL Server のサービスを外部に公開していない環境で，SQL Server のトラフィックを識別するといった場合に利用できる．

一方，アプリケーション層特有の攻撃を識別することはできず，またフラグメント化されたパケットを識別できないため，フラグメント化された不正トラフィックには対処できない．

2.3 まとめ

ルータをはじめとして，インターネットを構成するさまざまな機器においてパケット識別が行われている．さまざまな機器で実装されているパケット識別にはそれぞれ

表 2.7: NBAR のパケット識別

参照フィールド	送信元 IP アドレス，宛先 IP アドレス，トランスポート層 プロトコル，送信元 TCP/UDP ポート番号， 宛先 TCP/UDP ポート番号，ペイロードの一部
データの正規化	ネットワーク層，トランスポート層， アプリケーション層のデータの正規化が必要
フロー状態の保持	不明
必要な計算機資源	やや高い
不正トラフィックの識別	ベンダの定義したシグネチャによる

異なった目的がある．

本章では，不正トラフィックの識別を行う観点から，機構が参照するヘッダフィールド，データ正規化の有無，フロー状態の保持の有無，識別に必要となる計算機資源などを元に既存のパケット識別機構を評価し，不正トラフィックの識別のためのパケット識別機構の問題点を明らかにした．

第3章　関連研究

本章では，関連研究とその手法について述べる．

ネットワーク侵入検知においては，ベイズ推測，あるいはトラフィックの生起確率を用いた異常検知に関する先行研究が行われている．

3.1　ネットワーク侵入検知へのベイズ推測の応用

“ベイズ推論によるネットワーク侵入検知システムの開発”[8] は，侵入状態と侵入事象の因果関係グラフをであるベイズ信頼性ネットワーク (BBN) を用いて，侵入検知を行っている．BBN は，非常に実験的な取り組みであり，ホスト上での事象の変化を対象にしているため，ネットワークにそのまま応用することは難しい．特に，ホスト上での事象では，侵入状態と侵入事象の因果関係がある程度明らかであるため，ベイジアンネットワークを用いることが可能であるが，ネットワーク上ではトラフィックの状況と侵入状態の因果関係は必ずしも明らかではないため，ベイジアンネットワークを適用することは困難であると思われる．

PacketScore[9] は，DoS トラフィックを抑制する手法の研究である．被害ノードのトラフィックを元に統計データを作成，パケット単位による危険度をスコアとして算出し，DDoS 発生源近くで不正トラフィックを抑制する．本研究と PacketScore は，正常なトラフィックを元にプロファイルを作成し，パケット単位でのスコアリングを行うというアプローチが類似している．しかし，PacketScore が DoS の被害ノードを元にプロファイルを作成するのに対し，本研究ではハニーポットに対するトラフィックを元にプロファイルを作成する点，設定するパラメータが異なる．

ベイズ推測を用いたインターネットの脅威検知システムに関する研究 [10] は，IP パケットの到着間隔に基づいたベイズ推測を行うことにより，特定のポートに対する活動の増減を明らかにしている．この研究ではサービスポート番号ごとの脅威を求めているのに対して，本研究では，特定の IP プロトコルの生起確率に基づくベイズ推測により，パケット単位でスコアを算出しており，目的が異なる．

また，侵入検知そのものではなく，侵入検知のデータを元にベイズ推測を行うことで，興味深い結果を得ている研究に，“ベイズ推測を用いた不正侵入イベント増減予測”がある．これは，IDS において検知されたイベントの変更をもとにベイズ推測を行い，将来のイベント検知数の増減を予測する研究である．

[11] は，ベイズ推測によりトラフィックからアプリケーションの種類を推測している．

3.2 フロー情報を用いた侵入検知

また，侵入検知にフロー情報を活用する試みも行われている．

フローの特徴を用いた侵入検知に関する研究 [12] では，ネットワーク侵入者が非標準ポートや，標準的なポートを標準的ではない方法で利用する点に着目した侵入検知方法を提案している．この研究では，パケットフローをパケットのサイズ，到着間隔，送受信の比率，パケット間の相関関係を元に特徴付けることで，既知のフローに対して統計的なシグネチャを作成する．攻撃者は非標準ポートにバックドアを仕掛けたり，HTTP や Mail, DNS, ICMP などのプロトコルを用いてトンネルを仕掛けるため，未知のネットワークフローを特定することにより，侵入者のフローを特定するのに役立つ．

また，フローの特徴に基づくトラフィックの分類に関する研究 [13] では，機械学習に基づくアプリケーションフローの識別を提案している．

これらの研究では，フローの統計的なデータに基づき，トラフィックを識別している．しかし，トラフィックの識別にフローを用いているため，多くの計算機資源を必要とする．特にバックボーンネットワークにおいては，管理しなければならないフローの数が莫大となるため，管理が困難となる．

3.3 パケットヘッダによる侵入検知

PHAD[14] は，パケットのヘッダに対して異常検知を行うことで侵入を検知するシステムである．33 種類のヘッダフィールドをパラメータとして検査し，クラスタ化を行い，PPMC 法により異常スコアを算出する．PHAD では，不正トラフィックの識別において送信元ポート番号などの不正トラフィックに特異な傾向を有しないフィールドをパラメータとしているため，スコアを算出するコストが大きい．また，宛先ポート番号のクラスタ化を行っているため，特異な傾向を有するパラメータを識別する精度が低下している．

新たな異常検知の学習モデルである “非静的モデル” を提案した研究 [15] では，このモデルを適用した侵入検知システム “ALAD” を実装している．

異常検知の学習モデルでは，攻撃なしのトラフィックデータから学習を行い，検知中にパラメータを更新しない “静的モデル” と，攻撃を含むトラフィックデータから学習し，検知中にパラメータを更新する “動的モデル” が存在するが，前者はトラフィックの変化に対応できず，後者は攻撃を正常として学習するために誤検知を行う可能性が高いとしている．そのため，攻撃を含むトラフィックデータから学習を行い，テストデータから異常なイベントの発生間隔を求め，学習結果に反映されせ，異常スコアを割り当てる．

非静的モデルは，イベントの発生間隔が長い点に着目して異常検知を行っている．このため，非静的モデルは従来の IDS では検知しにくい不正アクセスを検知するのに有効であるとされている．しかし，現実のインターネットでは，必ずしも不正トラフィックのイベントの発生間隔が長いとはいえない．例えば，Slammer ワームや DoS などの

アタックについては，短い間隔でイベントが発生するため，イベントが一度発生した後においては有効に動作しないことが懸念される．

ALAD は，セッションとペイロードを対象に異常検知を行うシステムである．ALAD では，宛先 IP アドレス，また宛先 IP アドレスとポート番号ごとの送信元 IP アドレスの組み合わせ，あるいは宛先ポート番号ごとの TCP セッション確立・終了フラグ，ペイロードを学習させている．

これまで繰り返しの述べてきたように，ALAD はセッション管理とペイロードに対するパターンマッチに伴い，多くの計算機資源を必要するため，バックボーンネットワークへの適用は困難である．ただし，データの正規化を考慮していないため，今日の不正検知型 IDS/IPS ほどは計算機資源を必要としないと考えられる．

3.4 その他の関連研究

Cisco Guard XT[16] と Cisco Traffic Anomaly Detector は，Cisco Systems の (D)DoS 軽減アプライアンスと DDoS/DoS 検知アプリケーションである．Cisco Traffic Anomaly Detector で，通常のトラフィックパターンを学習させておき，DoS が発生した場合には，Cisco Guard XT が当該トラフィックを抑制する機能を持つ．提案手法は，正常トラフィックの学習を行う点が Cisco Guard XT と類似しているが，ハニーポットを用いる不適格なトラフィックを学習する点，またワームの検知も目的としている点が異なる．

第4章　パケットスコアリング手法の設計

本章では，提案手法の設計について述べる．

4.1　本研究のアプローチ

インターネット上を流れるトラフィックは，電子メール，ファイル転送，VoIP，電子商取引から，ワームの感染活動，DoS 攻撃，DDoS 攻撃，まさに玉石混淆である．そのため，ネットワーク資源を保護するには，ワームや DoS などの不適格トラフィックを判断する基準が必要である．

Firewall や IPS/IDS は，トラフィックを正当か不正を判断する二元論的アプローチを用いている．しかし，二元論的アプローチの判断には誤診断が伴い，未知のトラフィックに対する判断基準を策定する必要があることから，未知のワームや DDoS 攻撃等の異常なトラフィックからネットワークを保護するには不十分である．

また，現在の IPS/IDS や Firewall は，アプリケーション層やトランスポート層のデータを元に動作している前述した通り，現在の手法はスケーラビリティに乏しく，未知の攻撃への対応が困難である．

4.1.1　パケットベースアプローチ

次に，パケットベースのアプローチを説明する．既存の IPS/IDS は，データの正規化を行っているため，スケーラビリティや検知精度などの面で課題がある．本研究では，アプリケーションのセッションや TCP ストリームを構成する個々の IP パケットに着目し，一切ネットワーク層以上でのデータの正規化を行わないアプローチをとる．

例えば，通常のネットワークトラフィックにおいて，IP パケットがフラグメント化されることは稀である．筆者らがある研究・教育ネットワークにおいてトラフィックを観測したところ，フラグメント化されたパケットは 0.01 % 未満であった．フラグメント化されたパケットは，IDS 迂回攻撃に利用されることが知られており，リスクが通常よりも高いと考えることができる．

また，新規 TCP 接続を確立する際に発生する SYN フラグが有効なパケットは，平均的な TCP トラフィックよりもリスクが高いと考えることができる．これは，SYN Flooding 攻撃を受けた場合や，ワームによるサービスプローブを受けた場合には，SYN フラグが有効なパケットが相対的に高い割合となるからである．

スコアに基づくトラフィックコントロールを行うことにより，未知のワームの急激な感染拡大，あるいは DoS や DDoS 攻撃によるネットワークの帯域の輻輳からネットワークインフラストラクチャを保護できる．さらに，本手法は従来の Firewall や IPS といったトラフィック制御手法を併用することも可能である．

4.2 設計概要

提案手法の動作は，トラフィックプロファイルを作成する不正トラフィック収集・学習フェーズと，作成されたトラフィックプロファイルを用いてトラフィックをスコアリングするスコアリングフェーズの二段階に分類できる．

4.2.1 不正トラフィック収集・学習フェーズ

不正トラフィック収集・学習フェーズにおいては，ネットワークからトラフィックを収集することにより，ハニーポットのトラフィックプロファイルと，それ以外のトラフィックのプロファイルを構築する．ここでは，前者をハニーポットトラフィックプロファイル，後者を通常トラフィックプロファイルと呼ぶ．

図 4.1 に，不正トラフィックの収集・学習フェーズにおけるデータの流れを示す．

まず，プロファイル作成用のために，パケットキャプチャを行い，ネットワーク全体のトラフィックを収集する．次に，ハニーポットのトラフィックとそれ以外のトラフィックを分類するため，あらかじめハニーポットの IP アドレスを定義する．これにより，全収集トラフィックをハニーポットのトラフィックとそれ以外のトラフィックに分離する．

さらに，分離された各トラフィックからヘッダフィールドの値を抽出し，各トラフィックプロファイルに格納する．

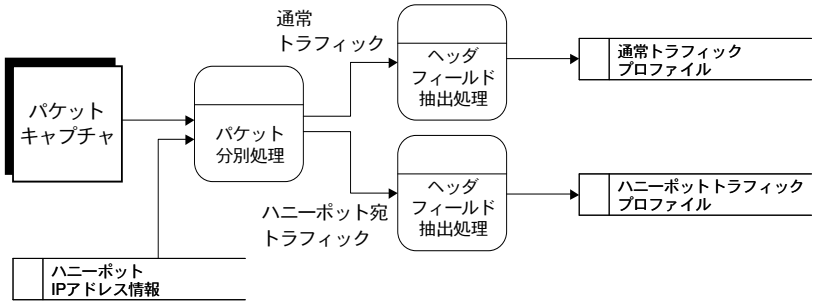

図 4.1: トラフィック収集・学習モジュール

4.2.2 スコアリングフェーズ

スコアリングフェーズにおいては，不正トラフィック収集・学習フェーズにおいて構築されたトラフィックプロファイルのデータに基づき，スコアリング対象トラフィック

のスコアリングを行う．

図 4.2 に，スコアリングフェーズにおけるデータの流れを示す．

スコアリングフェーズにおいては，スコアリング対象トラフィックの各ヘッダフィールドの値ごとに，通常トラフィックプロファイルとハニーポットトラフィックプロファイルの値を事前確率としたベイズ推測を行う．そして，各ヘッダフィールドの値をまとめ，パケットに対するスコアを算出する．

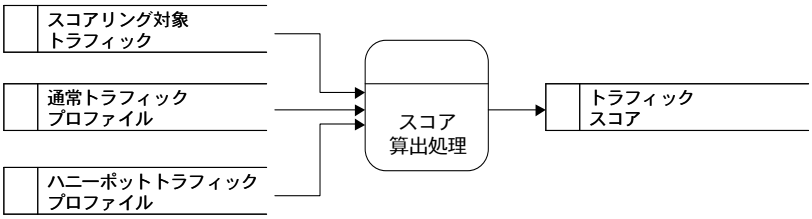

図 4.2: スコアリングモジュール

4.3 要件

4.3.1 網羅性

不正トラフィックを識別するにあたっては，いかなる種類のトラフィックも識別する必要があるため，不正パケット識別機構においては以下の事象に対応できる必要がある．

新たな攻撃への対応 セキュリティ情報サービスを提供している Symantec Corp. の調査 [17] によれば，2004 年 1 月 1 日〜6 月 30 日の間に，1,237 件の新たな脆弱性が発表されている．この件数は 1 日あたりおよそ 6.8 件の脆弱性が発表されている計算となる．また，新たな脆弱性が発見されてから攻略コードがリリースされるまでの平均日数は 6 日間となっており，日々新たな攻撃手法が登場しているといえる．

不正検知型の IDS/IPS は，攻撃に伴うトラフィックパターン，特にペイロードをシグネチャとして保持しておき，ネットワークトラフィックと照合することが一般的であるが，既知の攻撃以外は検知できない．

いわゆる 0-day exploit など，パッチがリリースされる以前の攻撃，あるいは脆弱性情報そのものが公開されないうちから攻撃に悪用されるケースが大きな問題となっており，不正トラフィックのクラス分け機構には，未知の攻撃手法にともなう不正トラフィックを検出できることが求められる．

フラグメント化されたトラフィックへの対応 トラフィックの識別機構においては，フラグメント化されたパケットも正しくクラス分けできることが必要である．

IP には，MTU のサイズを超過したパケットについては，複数のパケットに分割する IP fragmentation 機能がある．RFC1858[18] では，IP fragmentation を悪用した攻撃手法として，TCP ヘッダを意図的に分割するタイニー フラグメント攻撃，分割されたパケットの再構築処理により後から来たパケットのペイロードで当初のペイロードを上書きさせるオーバーラッピングフラグメント攻撃が行われる可能性が指摘されている．また，RFC3127[19] においても，これらの攻撃を組み合わせたタイニーオーバーラッピングフラグメント攻撃が行われる可能性が指摘されている．

IP fragmentation を悪用した攻撃を想定していないトラフィック識別機構では，不正なトラフィックを誤って正当なものとして分類する可能性がある．この結果，Firwall においてはアクセス制御リストの迂回される恐れがある．

TCP ストリームへの対応 IP ネットワークはベストエフォート型のネットワークであり，ネットワークの一部でパケットが失われるパケットロス，送信した順番と異なった順番で宛先に届くことに伴うリオーダリング，1 つのパケットが複数になる重複などの問題が発生する可能性がある．

このため，エンドノード，あるいは中継ノードのトランスポート層は，これらの問題に対処するためにデータ正規化処理を備えている必要がある．しかし，一部の TCP/IP 実装はこれらの処理が不完全なため，Firewall においてはアクセス制御リストの迂回，IDS/IPS においては検知回避攻撃 [6] が可能であることが指摘されている．

新プロトコルへの対応 インターネットにおいては，新たなネットワークアプリケーションのために，新しいアプリケーション層プロトコルがしばしば登場する．近年では，P2P アプリケーションで利用される Gnutella プロトコルや Napster プロトコル，Winny プロトコル，VoIP で利用される H.323 プロトコルや SIP(Session Initiation Protocol) などが登場している．

パケット識別機構は，これらの新たなプロトコルを適切に識別できることが必要である．

暗号化トラフィックへの対応 近年，従来は専用線などの保護されたリンク上で動作していたアプリケーションがインターネット上で利用されるようになった．これに伴い，VPN や電子商取引のような暗号技術が広く普及している．パケット識別機構にとって，一般に暗号化されたトラフィックの分類は一般的に困難である．特に，IDS/IPS などの既存手法においては，アプリケーション層のペイロードを元に識別を行っているが，ペイロードが暗号化されている場合には，不正トラフィックを検知することはできない．このため，SSL 経由で行われる攻撃や，暗号化されたボットネットの通信などを検知できることが望ましい．

4.3.2 スケーラビリティ

不正パケット識別には，以下の要件が必要である．

広帯域リンクへの対応 Gigabit Ethernet や 10Gbit Ethernet，あるいは OC-48,OC-192 などの高速インタフェースをサポートしたネットワーク機器の普及と低価格化により，広帯域ネットワークの利用が拡大している．特に，バックボーンネットワークは，複数のネットワークのトラフィックを集約している性質から，広帯域化が進んでいる．

そのため，パケット識別機構においても当然，広帯域ネットワークで利用されることを前提とした処理能力が必要である．

トラフィックフロー あるホストが同時に通信できる相手ホストの数には限りがあるが，ホストの数が多ければ多いほど，潜在的に多くの通信を同時に行うことができる．特に，バックボーンネットワークにおいては，多くのホストから発生するトラフィックを転送するため，存在するフローも莫大な数となる．

従って，不正トラフィックを識別するには，莫大なフローが存在することを前提とする必要がある．

4.3.3 精度

ネットワークインフラストラクチャを保護するための不正トラフィッククラス分け機構には，不正トラフィックを充分な精度で検出することが必要となる．充分な精度とは，100%の精度ではなく，バックボーンネットワークの輻輳など，可用性を損なう事態を防止するのに充分である割合のトラフィックを識別できることを意味する．

既存の IDS/IPS は，エンドノードの保護を目的としているため，すべての不正トラフィックを検出することが求められるが，本研究のトラフィック識別においては，全ネットワークトラフィックから，正常なトラフィックを除いたトラフィックを一定以上の量的割合で識別できることが求められる．

4.4 設計

本節では，IP ヘッダのフィールドをパラメータとして，個々の IP パケットをスコアリングする手法を提案する．

トラフィック収集モジュールでは IP パケットのヘッダからパラメータを抽出し，ベイズ推測によりトラフィックのプロファイルを作成する．そして，作成されたプロファイルに基づき，IP パケットの特性を判別する．

IP ヘッダ自体を対象にスコアリングを行う．これにより，多くの計算機資源を消費するトラフィックデータの正規化が不要となる．

4.5 トラフィック収集モジュール

本手法では，不正トラフィックのプロファイルと通常トラフィックのプロファイルを独立に学習させる．このため，トラフィック収集モジュールにおいては，収集した全収集トラフィックデータから不正トラフィックと通常トラフィックを分離できる必要がある．

本研究においては，トラフィック分離の課題を解決するため，ハニーポットの IP アドレスに基づく分離を行う．

ハニーポットに対するアクセスは全て不正アクセスであるため，全収集データからハニーポットの IP アドレスをもとにトラフィックを分離することで，不正トラフィックを取得できる．

通常トラフィックについては，全収集データから非ハニーポットホストに対するトラフィックを分離することにより取得できる．なお，通常トラフィックはトラフィックの一部に不正アクセスを含むこともあり得る．

4.6 設計のまとめ

本章においては不正パケット識別手法の設計を行った．本手法では，ハニーポットおよびバックボーンネットワークから，パケットのヘッダ情報を収集することで，ベイズ推測の学習データを取得する．この際，ハニーポットとバックボーンネットワークの学習データから，ベイズ推測を行う際に，有意なパラメータを取得することにより，本手法の実装に要する計算機資源を最小化にできる．

第5章　ハニーポットを用いた不正アクセスの検出

5.1　ハニーポット

5.1.1　ハニーポットの定義

Lance Spitzner の定義 [20] によれば，ハニーポットは次のように定義されている．

> “ハニーポットとは，資源に対する許可されていないアクセス，あるいは不正なアクセスを受けることによって価値が生じる情報システム資源である”．

ハニーポットと通常のシステムの違いは，システムが受けるアクセスの性質にある．通常のシステムは，何らかのサービスを提供する目的で設置されているのに対し，ハニーポットは正当な利用のためのサービスを提供しておらず，不正なアクセスを受ける目的で設置されるシステムである．従って，ハニーポットに対するアクセスを不正なアクセスと定義する．

5.1.2　ハニーポットの種類

ハニーポットは，攻撃者に対する対話性の程度により，低対話型と高対話型の二種類に分類できる．

低対話型ハニーポット

低対話型ハニーポットは，TCP/IP スタックやサービスのエミュレーションなどを通じて，攻撃者に対して限定的な対話性を提供するハニーポットである．低対話型のハニーポットには，次のような特徴がある．

●スケーラビリティに優れる　低対話型ハニーポットは，限定的な対話性のみを提供するため，単一のハードウェアで多数の仮想ホストを運用できる．

●運用リスクが低い　低対話型ハニーポットは，攻撃者にシステムへ侵入させることを想定していないため，ハニーポットを踏み台として外部のシステムに攻撃を行うことはない．従って，低対話型ハニーポットの運用リスクは低く，低い監視コストでハニーポットを運用できる．

●詳細なデータの収集が難しい　低対話型ハニーポットは，システムへの侵入を許さないため，侵入後の振る舞いを観察できない．また，容易にハニーポットがハニーポットであると気づかれる可能性がある．

低対話型ハニーポットの実装には，dumnet[21],KFSensor[22],Honeyd[23], SPECTER[24] などがある．

高対話型ハニーポット

高対話型ハニーポットは，実サービスを提供する通常のシステムとほぼ同様のシステムで構成することで，攻撃者に対して高い対話性を提供するハニーポットである．高対話型のハニーポットには，次のような特徴がある．

●詳細なデータの収集が可能　高対話型ハニーポットは，実システムを用意し，攻撃者にシステムに侵入させ，可能な限り多くの行動を許容するシステムである．従って，侵入後の振る舞いを観察することができるため，攻撃手法などの詳細なデータを収集できる．また，低対話型ハニーポットと比較した場合，ハニーポットがハニーポットであると気づかれにくい．

●スケーラビリティに劣る　高対話型ハニーポットの実装には実システムが必要になるため，単一のハードウェアで単一，あるいは少数のハニーポットの運用に限定される．従って多数のホストを配置する必要がある用途に高対話型ハニーポットを用意するのは困難である．

●運用リスクが高い　高対話型ハニーポットには，攻撃者にシステムへの侵入を許すため，複数のリスクがある．第一のリスクはハニーポットを踏み台として外部のシステムに攻撃を行うリスクである．このため，外部のシステムに対して攻撃を禁止する，あるいは一定程度までしか許可しないよう，監視を行うために高い監視コストが必要となる．第二のリスクは，ハニーポット機能を無効化されるリスクである．高対話型ハニーポットは通常，実システムに監視機能などを追加して構築されるが，これらの機能が無効化されることでハニーポットとして機能しなくなるリスクがある．第三のリスクは，不正利用である．高対話型ハニーポットは，高い自由度を有するため，ハニーポットが海賊版のソフトウェアや音楽，映画のアーカイブサイトとして悪用される恐れがある．また，これらの不正行為が行われた場合，システムの管理責任を問われる可能性がある．

高対話型ハニーポットの実装には，Symantec Decoy Server[25], Honeynets[26] などがある．

5.1.3 ハニーポットの選択

ハニーポットに対する要件

本研究で用いるハニーポットには，次のような要件がある．

スケーラビリティ より多くの不正なアクセスに伴うトラフィックを収集するためには，ハニーポットが不正なアクセスを受ける確率を向上させる必要がある．ただし，ハニーポットは，それ自体が正体を明らかにするとハニーポットとしての価値が大きく損なわれる．従って，ハニーポットの IP アドレスに対してアクセスを誘導することは難しい．そこで，本研究ではハニーポットに多数の IP アドレスを割り当てることで，不正なアクセスを受ける確率を高める．従って，本研究で用いるハニーポットには多数の IP アドレスを利用できるスケーラビリティが必要である．

運用リスク 本研究の目的は，ワームや (D)DoS などの不正トラフィックの識別である．前項で述べたように，ハニーポットの運用には様々な種類のリスクが存在するが，できるだけ低いリスクで目的を達成できることが望ましい．従って，踏み台となる可能性が低いこと，ハニーポット機能が無効化されにくいこと，ハニーポットの不正利用に伴う問題が発生しないことが必要である．

本研究では，以上の要件を満たすため，本研究ではトラフィック収集用のハニーポットとして低対話型ハニーポットを用いる．

ハニーポットの稼働レイヤ

低対話型ハニーポットは，エミュレーションを行うレイヤによって，さらにトランスポート層型とアプリケーション層型の二種類に分類できる．トランスポート層型のハニーポットは，全 TCP ポート番号宛のトラフィックに対して包括的なエミュレーションを提供できる．

一方，アプリケーション層型のハニーポットは，特定の TCP ポート番号宛のトラフィックにしか対応できないが，サービスの種類に応じたエミュレーションを提供できる．

本研究では，不正パケットのスコアリングに適した低対話型ハニーポットを検討するため，トランスポート層型の Dumnet とアプリケーション層型の KFSensor の両方をほぼ同一条件で運用し，それらのトラフィックの傾向と，それらのトラフィックに基づいたスコアリングの評価を行う．

Dumnet

Dumnet は，トランスポート層でエミュレーションを行う低対話型ハニーポットで，UNIX 系 OS 上で稼働する．Dumnet は，次の機能により，あらかじめ定義した IP アド

表 5.1: Dumnet の運用環境

ハードウェア	DELL PowerEdge 1750
CPU	Intel Xeon 2.4GHz
HDD	73GB (10,000rpm)
メモリ	512MB
OS	Debian GNU/Linux 3.1 (kernel 2.4.27)
ハニーポットソフトウェア	dumnet-1.4

レス空間にホストが存在するように振る舞う．表 5.1 に，今回の実験における Dumnet の運用環境を示す．

ICMP エミュレーション機能 指定された IP アドレス宛の ICMP ECHO Request メッセージに対して，ECHO Reply メッセージを応答する．

TCP エミュレーション機能 SYN フラグがセットされた TCP パケットに対して，SYN+ACK フラグをセットしたパケットを応答することで，スリーウェイハンドシェイクを成立させ，全ての TCP ポートが開かれているかのように動作する．

Dumnet は，インタフェースに対して bind() を行うのではなく，libpcap を用いてパケットを取得し，独自の TCP スタックによる TCP パケットを処理するため，スケーラビリティに優れ，多くの IP アドレスに対するエミュレーションを提供できる．

KFSensor

KFSensor は，アプリケーション層でエミュレーションを行う低対話型ハニーポットで，Microsoft Windows で稼働する．KFSensor の特徴は，CIFS(Common Internet File System) サーバや IIS のエミュレーションなど，Windows のエミュレーションに優れている点にある．

実験では，表 5.2，表 5.3，表 5.4，表 5.5 に示したポートにおいて，KFSensor がアプリケーションのエミュレートなどを行う構成とした．

また，KFSensor の設定を表 5.6 に，KFSensor を運用環境を表 5.7 に示す．

表 5.2: Main Scenario で定義したポート (通常の TCP サービス)

ポート番号	サービス	動作
21/tcp	FTP	エミュレーション (GuildFTPd)
25/tcp	SMTP	エミュレーション (Microsoft ESMTP MAIL Service)
42/tcp	WINS	受信・切断
53/tcp	DNS	受信・切断
68/tcp	DHCP	受信・切断
80/tcp	HTTP	エミュレーション (Microsoft-IIS/6.0)
81/tcp	HTTP	エミュレーション (Microsoft-IIS/6.0)
82/tcp	HTTP	エミュレーション (Microsoft-IIS/6.0)
83/tcp	HTTP	エミュレーション (Microsoft-IIS/6.0)
88/tcp	Kerberos	受信・切断
110/tcp	POP3	エミュレーション (Microsoft Windows POP3 Service Version 2.0)
119/tcp	NNTP	受信・切断
135/tcp	MS RPC	エミュレーション
139/tcp	NBT Session Service	エミュレーション
389/tcp	LDAP	受信・切断
443/tcp	HTTPS	受信・切断
445/tcp	NBT SMB	エミュレーション
464/tcp	kpasswd	受信・切断
522/tcp	NetMeeting User Location	受信・切断
563/tcp	NNTP SSL	受信・切断
593/tcp	RPC over HTTP	エミュレーション
636/tcp	LDAP SSL	受信・切断
1025/tcp	MS RPC 1025	受信・切断
1026/tcp	MS RPC 1026	受信・切断
1027/tcp	MS RPC 1027	受信・切断
1028/tcp	MS COM+ Internet Services	エミュレーション
1080/tcp	WinGate	受信・切断
1214/tcp	Grokster, P2P file sharing	受信・切断
1433/tcp	SQL Server	エミュレーション
1494/tcp	Citrix	受信・切断
2234/tcp	Directplay	受信・切断
3128/tcp	IIS Proxy	エミュレーション (Microsoft-IIS/6.0)
3268/tcp	Global Catalog Service	受信・切断
3389/tcp	Windows Terminal Server	エミュレーション
4662/tcp	eDonkey2000, P2P file sharing	受信・切断
5000/tcp	MS Universal Plug and Play	受信・切断
5631/tcp	PC Anywhere 1	受信・切断
5632/tcp	PC Anywhere 2	受信・切断
5900/tcp	VNC	エミュレーション
6112/tcp	CDE	受信・切断
6346/tcp	BearShare	受信・切断
6881/tcp	BitTorrent	受信・切断
8080/tcp	IIS Proxy	エミュレーション (Microsoft-IIS/6.0)

表 5.3: Main Scenario で定義したポート (通常の UDP サービス)

ポート番号	サービス	動作
42/udp	WINS	切断
88/udp	Kerberos	受信・切断
137/udp	NBT Name Service	エミュレーション
138/udp	NBT Datagram Service	エミュレーション
161/udp	SNMP	受信・切断
389/udp	LDAP	受信・切断
1434/udp	SQL UDP Server	エミュレーション
1900/udp	MS Universal Plug and Play	受信・終了
3268/udp	Global Catalog Service	受信・終了
4672/udp	eMule	受信・終了

表 5.4: Main Scenario で定義したポート (TCP ベースのトロイの木馬およびバックドア類)

ポート番号	サービス	動作
999/tcp	WinSatan	受信・切断
1024/tcp	NetSpy, Trojan	受信・切断
2023/tcp	Hack City Ripper Pro, Trojan	受信・切断
2774/tcp	SubSeven Chat	エミュレーション (kfSubSeven Chat)
3127/tcp	MyDoom Backdoor	エミュレーション (MyDoom)
3355/tcp	Hogle SMTP, Trojan	エミュレーション (SMTP)
3410/tcp	OptixPro, Trojan	受信・切断
4444/tcp	Blaster, Trojan	エミュレーション (Command console)
5554/tcp	Sasser	エミュレーション (Command console)
6129/tcp	Dameware	エミュレーション (Dameware)
6912/tcp	Shit Heep, Trojan	切断
6969/tcp	GateCrasher, Trojan	受信・切断
6970/tcp	GateCrasher, Trojan	受信・切断
7394/tcp	Unknown	受信・切断
7215/tcp	SubSeven Matrix	エミュレーション (kfSubSeven Matrix)
8879/tcp	Hack Office Armageddon	受信・切断
8967/tcp	Command console	エミュレーション (Command console)
9898/tcp	Command console	エミュレーション (Command console)
9996/tcp	Sasser worm conole	受信・切断
10085/tcp	Syphillis, Trojan	切断
11768/tcp	dipnet, trojan	受信・切断
17300/tcp	Kuang 2, Trojan	受信・切断
20000/tcp	Millennium, Trojan	切断
20034/tcp	NetBus	受信・切断
20168/tcp	Lovegate worm	エミュレーション (Command console)
20525/tcp	Unknown	受信・切断
21544/tcp	GirlFriend, Trojan	受信・切断
23476/tcp	Donald Dick	切断
27015/tcp	Half Life	受信・切断
27374/tcp	SubSeven Server	エミュレーション (kfSubSeven Server)
29891/tcp	The Unexplained, Trojan	受信・切断
31337/tcp	Eleet, Trojans	受信・切断
36794/tcp	Bugbear	受信・切断
41170/tcp	Unknown	受信・切断
54283/tcp	SubSeven2.1 Trojan	エミュレーション (SubSeven2.1 Trojan)
54320/tcp	Back Orifice 2000, Trojan	受信・切断
57341/tcp	NetRaider, Trojan	受信・切断
60609/tcp	mIRC Trojan	受信・切断
65432/tcp	The Traitor, Trojan	受信・切断
65535/tcp	RC1 Trojan, Trojan	受信・切断

表 5.5: Main Scenario で定義したポート (UDP ベースのトロイの木馬およびバックドア類)

ポート番号	サービス	動作
69/udp	Blaster, Trojan	受信・切断
10000/udp	Unknown	受信・切断
20000/udp	Millennium, Trojan	受信・切断
29891/udp	The Unexplained, Trojan	受信・切断
31337/udp	Back Orifice, Trojan	切断
65432/udp	The Traitor, Trojan	受信・切断

表 5.6: KFSensor の設定パラメータ

設定パラメータ	値
MaxClients	256
MaxConcurrentConnectionsPerIP	20
MaxConnectionsPerIP	100
MaxPortsPerIP	100
MaxReceiveSize	1024000
MaxTCPConnections	1000000
MaxUDPConnections	1000000

表 5.7: KFSensor の運用環境

ハードウェア	SuperMicro 6022C
CPU	Intel Xeon 2GHz * 2
HDD	73GB * 3
メモリ	2GB
OS 仮想化ソフトウェア	VMware GSX Server 3.1.0
割り当て HDD	10GB
割り当てメモリ	384MB
ゲスト OS	Windows Server 2003 Enterprise Edition
ハニーポットソフトウェア	KFSensor Enterprise Edition Version 3.0.4

第6章　トラフィックの分析

本章では，ハニーポットを用いた実験と，実験において収集されたトラフィックの分析結果について述べる．

6.1　実験の概要

6.1.1　ハニーポットと IP アドレス空間

ハニーポットは不正アクセスを受ける目的で設置されるシステムであるため，正当な利用目的で通常のシステムと誤ってアクセスされることを防ぐ必要がある．

本研究では，ハニーポット専用の IP アドレス空間と研究・教育用ネットワーク (R&E Network) で利用されている IP アドレス空間の二種類の IP アドレス空間を用いてハニーポットを運用した．

第一の IP アドレス空間は，それまで未使用だったクラス B の IP アドレス空間を，ハニーポットの運用開始と同時にインターネットに広報し，ハニーポット専用のアドレス空間として用いた．便宜上，本論文ではこの IP アドレス空間を WIDE133 と呼ぶ．WIDE133 には，通常のシステムは存在せず，DNS の正引きは設定されていない．

V. Yegneswaran らの研究 [27] によれば，未使用 IP アドレスブロックを用いたトラフィック観測においては，主に次のようなパケットが観測されたことが報告されている．

- Backscatter (送信元 IP アドレスをパケットに対する応答)
- ホストのプローブ
- ワームの感染活動に伴うパケット
- SMTP サーバに対するスキャン

筆者らも 2004 年より低対話型ハニーポットを運用しているが，ハニーポットで収集したトラフィックにも同様の傾向が見られ，正当な利用目的と思われるトラフィックはほとんど確認できなかった [28].

第二の IP アドレス空間は，ある研究・教育用ネットワークが利用している IP アドレス空間から一部の IP アドレスをハニーポット用に割り当てた空間である．このネットワークは，表 6.1 および表 6.2 のサブネットで構成される．便宜上，本研究では研究・教育用ネットワークのサブネット群を RENET と呼ぶ.

表 6.1: Firewall を経由せず直接インターネットと接続しているサブネット

サブネットのサイズ	ネットワークの呼称
/24	RENET302:1
/25	RENET302:2

表 6.2: Firewall を経由してインターネットと接続しているサブネット

サブネットのサイズ	ネットワークの呼称
/25	RENET303
/25	RENET307
/27	RENET304
/27	RENET305

実験では RENET302:1，RENET303，RENET307 の各サブネットより 20 アドレスずつ，合計 60 アドレスをハニーポット全体に割り当てた．各サブネットでは DHCP による動的 IP アドレス割り当てを行っており，実験開始前の時点で DHCP プールに含まれていた IP アドレスをハニーポットに割り当てている．このため，固定 IP アドレスで提供されているサービスに対してアクセスが行われる可能性は低い．

いずれの場合も，従って通常のシステムに対する正当なアクセスのつもりで，誤ってハニーポットに対してアクセスが行われる可能性はほとんどない．このため，本研究においてはハニーポット宛のトラフィックを全て不正トラフィック，ハニーポットを含まない対照ネットワークのトラフィックを通常トラフィックと定義する．

6.1.2 ネットワークの構成

ハニーポットのトラフィックと，比較対照データである研究ネットワークのトラフィックを収集した．図 6.1 に実験に用いたネットワーク環境の概要を示す．

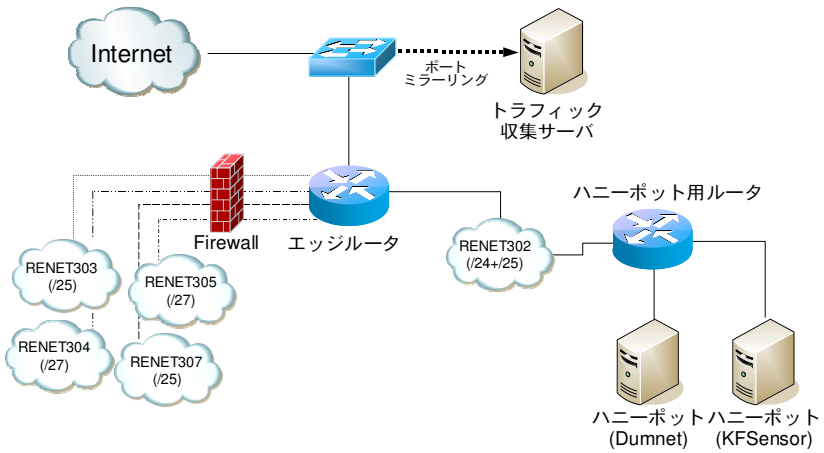

図 6.1: 実験に用いたネットワークトポロジ

6.1.3 ハニーポットと IP アドレスの割り当て

本実験では，稼働レイヤの異なるハニーポット宛のトラフィックの差異と異なる IP アドレス空間におけるトラフィックの差異を明らかにするため，表 6.3 の組み合わせでハニーポットを構成した．また，図 6.2 の通り，ルーティングを設定した．

表 6.3: ハニーポットの構成

名称	IP アドレスブロック	ハニーポット
研究・教育用ネットワーク	RENET(ハニーポットなど[1]を除く)	N/A
Dumnet/WIDE133	WIDE133 (/16)	Dumnet
Dumnet/RENET	RENET302:1, RENET303, RENET307 の各サブネットより各 10 ホスト，30 ホスト	Dumnet
KFSensor	RENET302:1, RENET303, RENET307 各サブネットより各 10 ホスト，30 ホスト	KFSensor

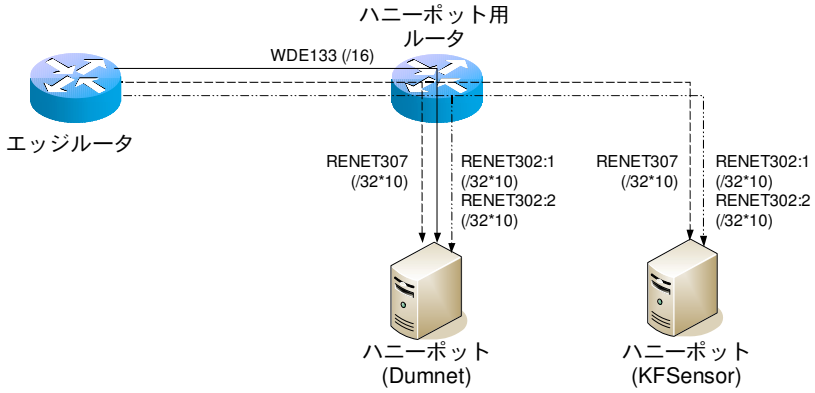

図 6.2: 実験に用いた IP アドレスに対するルーティングの設定

6.1.4 トラフィックの収集

実験ネットワークの上流に位置する L2 スイッチにおいて，実験ネットワーク宛のパケットをトラフィック収集サーバにコピーするようポートミラーを設定した．トラフィック収集サーバにおいては，WIDE133 および RENET 宛のパケットを一括して保存し，オフラインで宛先 IP アドレスに基づいてトラフィックを分離した．

6.1.5 トラフィック収集期間

本研究では，事前に与えられたトラフィックをパラメータとして，新たなパケットのスコアリングを行う．本実験では，2005 年 12 月 27 日 (火)0 時 00 分〜2005 年 12 月 27 日 (火)23 時 59 分にパラメータとして用いるトラフィックを収集した．なお，特段の断りがない限り，全ての時刻は日本時間 (JST) で表記した．

[1]RENET には，Linux ディストリビューションや Web ブラウザのミラーサイトをしている Anonymous FTP サーバが存在するが，通常のホストとは異なった性質を持つサーバであるため，当該 IP アドレス宛のトラフィックは統計から除外している．以下の RENET においても同様である．

6.1.6 IP ヘッダの各フィールドの傾向

本節では，低対話型ハニーポット，および研究・教育用ネットワークに送信されたトラフィックの IP ヘッダの各フィールドを分析結果を示す．各ヘッダフィールドにおける傾向を把握することで，パケットのパケットスコアリング手法に適したパラメータを決定する．

2005 年 12 月に収集した各ハニーポットのトラフィックの概況を表 6.4 に，またパケットスコアリングを行う対象のトラフィックの概況を表 6.5 に示す．

なお，収集したトラフィックデータは，インターネットからハニーポットに送信された受信側のみを収集しており，ハニーポットからインターネット宛に送信されるトラフィックは含まれない．

表 6.4: 収集トラフィックの概略 (2005 年 12 月)

	研究・教育用ネットワーク[2]	Dumnet/ WIDE133	Dumnet/ RENET	KFSensor
IP アドレス数	631[3]	65,534	30	30
通信先ホスト数	160,093	62,175	947	1,355
1IP アドレスあたりの通信先ホスト数	253.71	0.95	31.57	45.17
受信パケット数	22,169,190	12,568,731	213,226	64,659,985
平均スループット (pps)	51.32	145.47	2.46	748.38
1IP アドレスあたりの受信パケット数	35,133	192	7,107	2,155,332
受信トラフィック (KBytes)	8,002,699	1,081,915	13,405	8,982,633
平均スループット (byte/s)	92,624	12,522	155	103,965
1IP アドレスあたりの受信トラフィック (KBytes)	12,683	16	447	299,421

計測時間:2005 年 12 月 27 日 (火)0 時 00 分〜23 時 59 分

表 6.5: 収集トラフィックの概略 (評価対象)

	研究・教育用ネットワーク[2]	Dumnet/ WIDE133	Dumnet/ RENET	KFSensor
IP アドレス数	631[3]	65,534	30	30
通信先ホスト数	190,942	127,525	750	1,133
1IP アドレスあたりの通信先ホスト数	302.60	1.95	25.00	37.77
受信パケット数	19,351,235	81,262,948	195,826	36,921,604
平均スループット (pps)	223.97	940.54	2.27	427.33
1IP アドレスあたりの受信パケット数	30,668	1,240	6,528	1,230,720
受信トラフィック (KBytes)	4,831,498	2,211,212	12,655	11,871,318
平均スループット (byte/s)	55,920	25,593	146	137,400
1IP アドレスあたりの受信トラフィック	7,657	34	422	73,707

計測時間:2005 年 12 月 29 日 (木)0 時 00 分〜23 時 59 分

[2]研究・教育用ネットワークのデータについては，実験環境の制約により全トラフィックを収集できなかったため，5:1 の割合でサンプリングを行った．通信先ホスト数はサンプリング後の観測値，その他の項目については観測値を 5 倍した推定値を示す．

[3]ネットワークアドレスとブロードキャストアドレス，Anonymous FTP サーバのアドレスを除く

6.2 各トラフィックにおける IP ヘッダの値の傾向

6.2.1 トランスポート層プロトコルの構成比率 (2005 年 12 月)

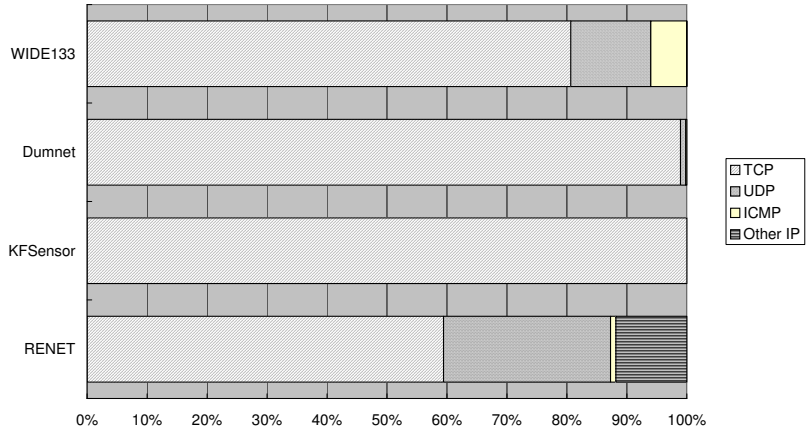

図 6.3: 各トラフィックにおけるプロトコルあたりのパケット数の割合

各パケットのヘッダフィールドから，各トラフィックにおけるトランスポート層プロトコルの構成比率を図 6.3 および表 6.6 に示す．なお，小数点以下 3 位で四捨五入を行ったため，合計は必ずしも 100%とはならない．

表 6.6: トランスポート層プロトコルの構成比率 (2005 年 12 月)

	研究・教育用ネットワーク[4]	Dumnet/WIDE133	Dumnet/RENET	KFSensor
TCP	59.44%	98.95%	80.64%	100.0%
UDP	27.84%	0.85%	13.35%	0.0%
ICMP	0.89%	0.20%	5.98%	0.0%
その他の IP	11.84%	0.0%	0.03%	0.0%

表 6.6 からは，研究・教育用ネットワークとハニーポットのトラフィックでは，トランスポート層プロトコルの構成比率が異なることがわかる．特に，前者においては，IPv6 over IPv4 トンネルに伴うトラフィックが発生しているため，“その他の IP” の項目に含まれるトラフィックの割合が他よりも高い．また，Skype 等のアプリケーションの利用により，UDP トラフィックに占める割合も高い．

また，同一のハニーポット実装を用いている Dumnet/WIDE133 と Dumnet/RENET のトラフィックを比較することで，異なる IP アドレス空間におけるトラフィック傾向の違いを把握できる．Dumnet/WIDE133 と Dumnet/RENET では，トランスポート層プロトコルの構成比率が異なっている一方，Dumnet/RENET と研究・教育用ネットワークのそれは類似している．

これらのことから，通常トラフィックに近いプロトコル構成比率を得るには，同一 IP アドレス空間において Dumnet を稼働させるのがよいといえる．

6.2.2 IP パケットのフラグメンテーションビットの割合

続いて，フラグメント化した IP パケットの割合を調査した．IP パケットがフラグメント化しているかは IPv4 パケットヘッダに含まれるフラグメンテーションビットによって判断できる．

図 6.4 および表 6.7 に各トラフィックにおけるフラグメンテーションビット値とパケット数の割合を示す．

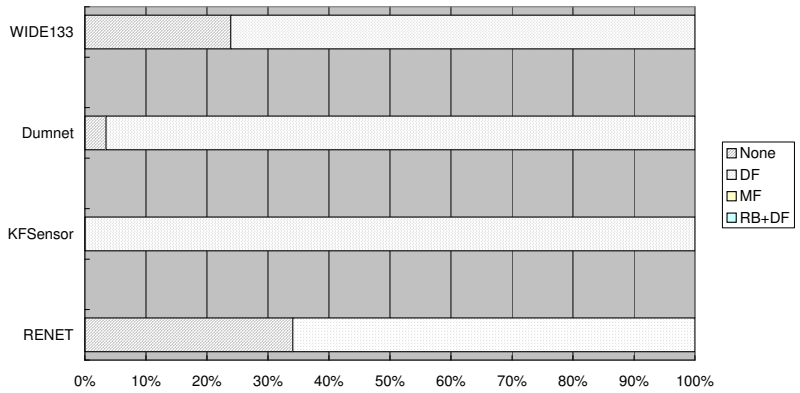

図 6.4: 各トラフィックにおけるフラグメンテーションビット値とパケット数の割合

表 6.7: フラグメンテーションビット値の構成比率 (2005 年 12 月)

	研究・教育用ネットワーク	Dumnet/WIDE133	Dumnet/RENET	KFSensor
なし	34.08%	23.91%	3.46%	0.03%
DF ビット	65.92 %	76.09%	96.54 %	99.97%
MF ビット	0.0%	0.0%	0.0%	0.0%
予約ビット+DF ビット	0.0%	0.0%	0.0%	0.0%

いずれのトラフィックにおいても DF ビットがセットされていないトラフィックが過半数を占めるが，同一 IP アドレス空間を用いている Dumnet/RENET，KFSensor，RENET の中でも，RENET では DF ビットがセットされていないパケットの割合が 34.08%ともっとも高い．

6.2.3 IP パケットの TTL フィールドの傾向

IPv4[29] ヘッダに含まれる TTL フィールドは，パケットの生存期間を定義した 8bit の値である．

2005 年 12 月に観測された各トラフィックにおける TTL あたりのパケット数の割合を図 6.5，図 6.6，図 6.7，図 6.8 に示す．

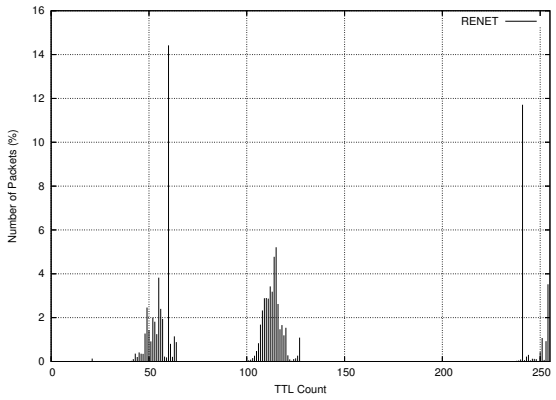

図 6.5: RENET における TTL の割合

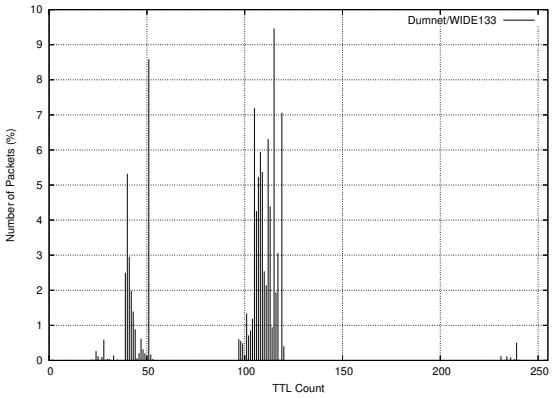

図 6.6: Dumnet/WIDE133 における TTL の割合

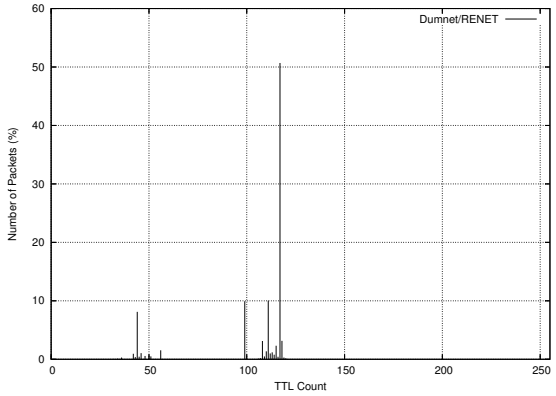

図 6.7: Dumnet/RENET における TTL の割合

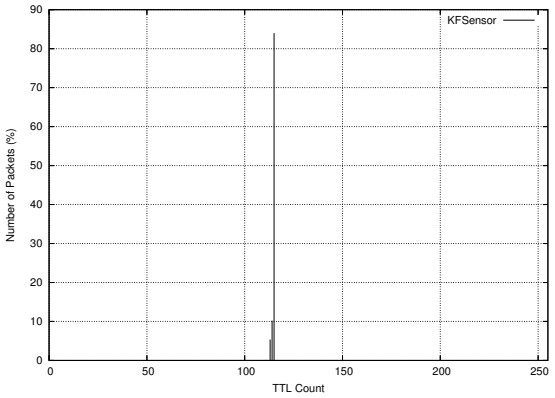

図 6.8: KFSensor における TTL の割合

観測 IP アドレス数の多い，図 6.5，図 6.6 においては，TTL 値が 50 前後，110 前後，250 前後の三点において大きな割合を占めていることがわかる．

一方，観測 IP アドレス数の少ない，図 6.7 では TTL 値 117，図 6.8 では TTL 値 115 と，特定の TTL 値が多くの割合を占めている．

6.2.4 IP パケットのホップ数の傾向

TTL は，TCP/IP スタックの初期 TTL に基づいた値をとるが，OS によって初期 TTL の値が異なるため，数値尺度として数の大小を比較するのに適さない．そこで，パケット受信時の TTL の値と一般的な OS の初期 TTL との差分を求めることで，送信元ホストからのホップ数を推定できる [30]．本実験においては，ntop[31] のアルゴリズムに基づき，推定ホップ数を算出した．

表 6.8: 推定ホップ数の算出表

TTL	推定ホップ数
0	0
1〜8	TTL 値 - 1
9〜32	32 - TTL 値
33〜64	64 - TTL 値
65〜128	128 - TTL 値
129〜255	255 - TTL 値

2005 年 12 月に観測された各トラフィックにおける推定ホップ数あたりのパケット数の割合を図 6.9，図 6.10，図 6.11，図 6.12 示す．

これらのグラフからは，ハニーポットと通信したホストの推定ホップ数の分布と研究・教育用ネットワークの推定ホップ数の分布に違いが見られる．

この理由として，研究・教育用ネットワークは日本国内など，ホップ数の少ないホストとの通信が多いのに対し，ハニーポットでは研究ネットワークよりも国外などのホップ数の多いホストとの通信が多いことが推察される．

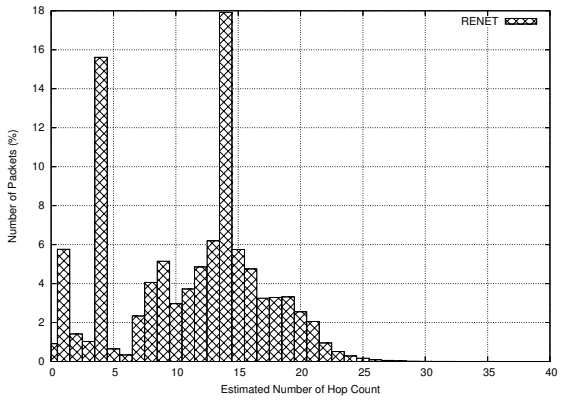

図 6.9: RENET の推定ホップ数の割合

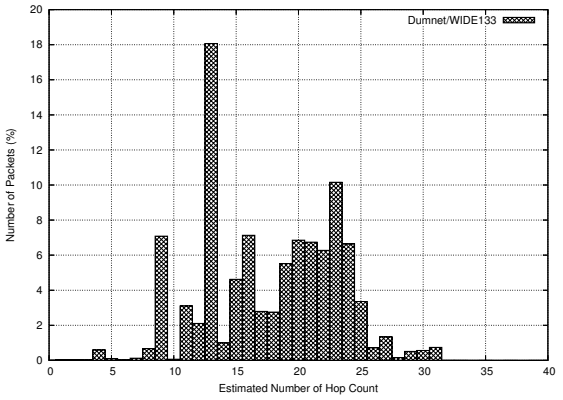

図 6.10: Dumnet/WIDE133 の推定ホップ数の割合

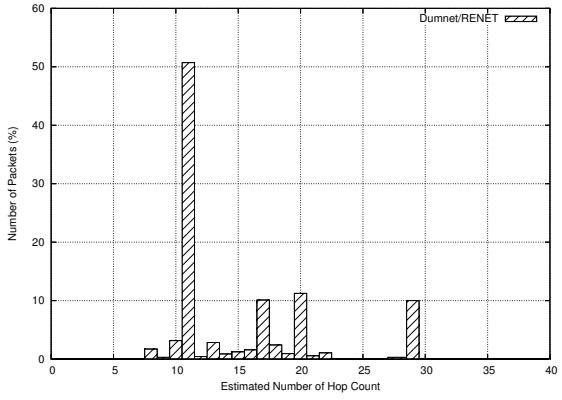

図 6.11: Dumnet/RENET の推定ホップ数の割合

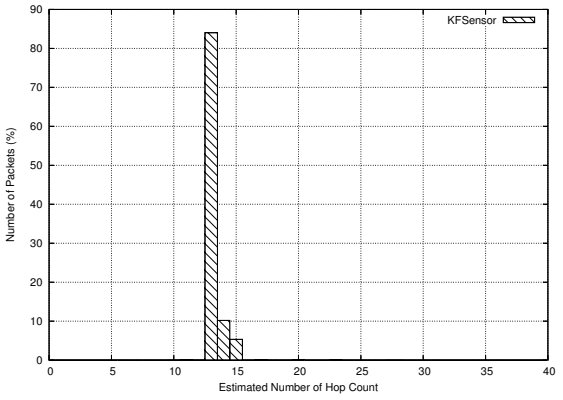

図 6.12: KFSensor の推定ホップ数の割合

6.2.5 IP パケットのデータグラム長の分布

データグラム長は，IPv4 ヘッダを含むパケット長を表現する 16bit のヘッダフィールドである．

2005 年 12 月に観測された各トラフィックにおけるデータグラム長あたりのパケット数の割合を図 6.13，図 6.14，図 6.15，図 6.16 に示す．

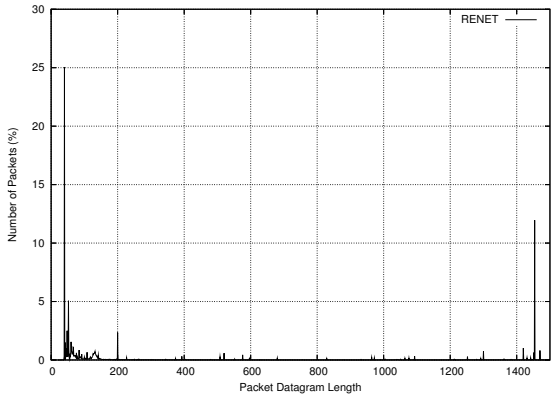

図 6.13: RENET のデータグラム長の分布

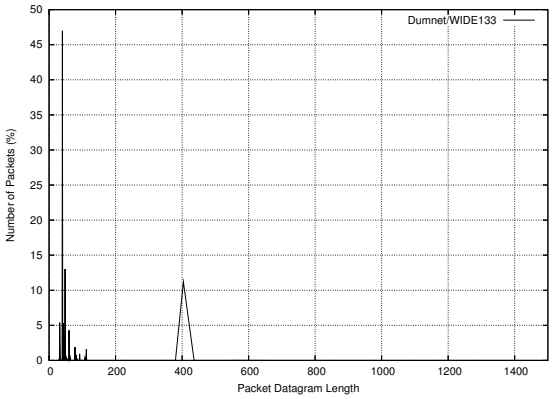

図 6.14: Dumnet/WIDE133 のデータグラム長の分布

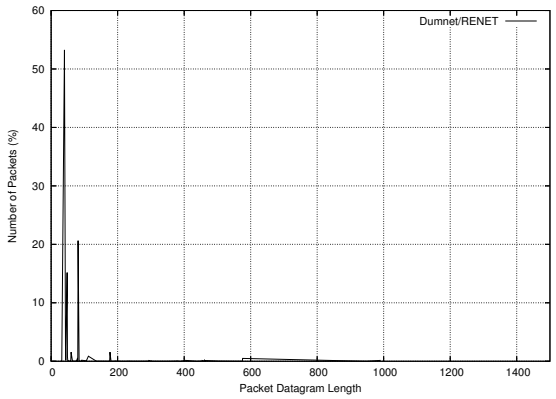

図 6.15: Dumnet/RENET のデータグラム長の分布

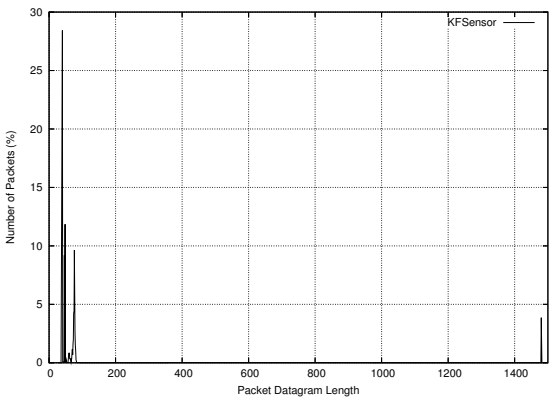

図 6.16: KFSensor のデータグラム長の分布

研究・教育ネットワークとハニーポットでは，パケット長の分布に違いがある．研究・教育ネットワークでは，48 バイト付近のパケット長に分布しているのに加え，1454 バイト付近にも分布が見られる．一方，各ハニーポットにおいては，ほとんどが 48 バイト付近のパケット長に分布しており，KFSensor を除けば 1454 バイト付近には分布が見られない．これは，ハニーポット宛てのパケットには，パケット長の短いスキャンやサービスプローブ行為が多く含まれる影響と考えられる．

6.2.6 IP パケットの ToS フィールド値の割合

ToS(Type Of Service) フィールドは，IP パケットの処理を IPv4 ヘッダにおいて指定するヘッダフィールドである．

図 6.17 に各トラフィックにおける ToS フィールド値とパケット数の割合を示す．

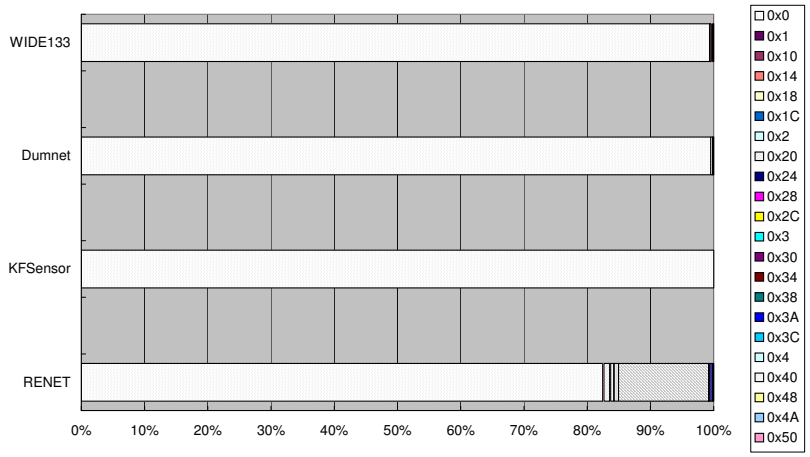

図 6.17: 各トラフィックにおける ToS(Type of Service) 値とパケット数の割合

ToS はトラフィックの種別を定義した 8 ビットの IP ヘッダフィールドである．すべてのトラフィックにおいて，ToS フィールドの値は 0x0 であるが，研究・教育用ネットワークにおいては，0x0 以外の値をとるトラフィックが 17.51%を占めている．このため，ToS フィールドの値が 0x0 以外の値であるトラフィックは，非不正トラフィックである可能性がある．

6.2.7 TCP 宛先ポート番号あたりのパケット数の割合

2005 年 12 月に観測された各トラフィックにおける TCP 宛先ポート番号あたりのパケット数の割合を図 6.18，図 6.19，図 6.20，図 6.21 に示す.

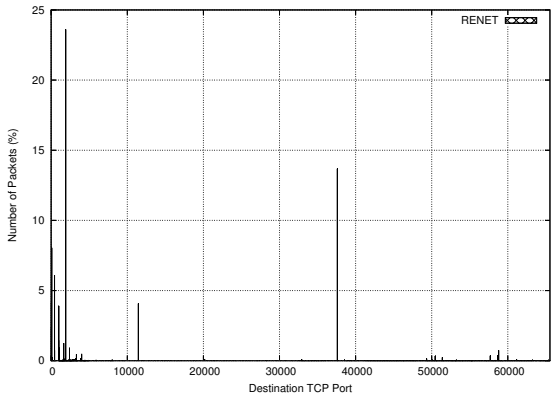

図 6.18: RENET における宛先 TCP ポート番号の分布

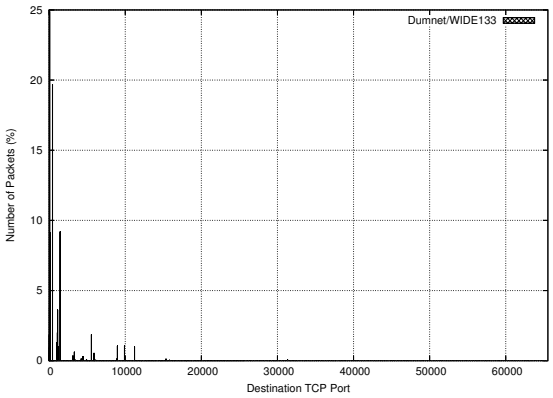

図 6.19: Dumnet/WIDE133 における宛先 TCP ポート番号の分布

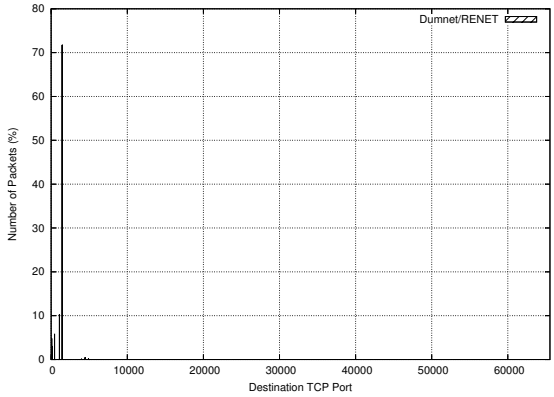

図 6.20: Dumnet/RENET における宛先 TCP ポート番号の分布

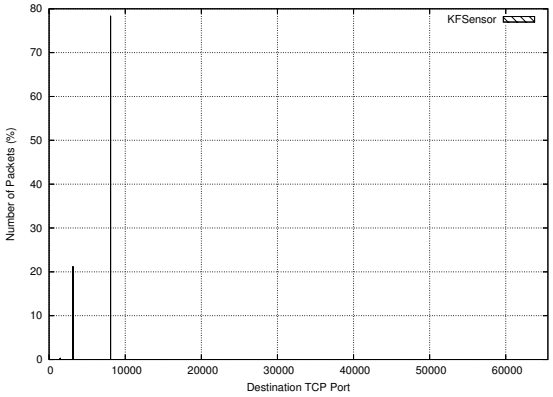

図 6.21: KFSensor における宛先 TCP ポート番号の分布

これらの結果からは，観測 IP アドレス数の多い，図 6.18，図 6.19 においては，多くのポートに対してアクセスが分散しているのに対し，観測 IP アドレス数の少ない，図 6.20，図 6.21 では特定のポートに対するアクセスが多くの割合を占めていることがわかる.

6.2.8 TCP 送信元ポート番号あたりのパケット数の割合

2005 年 12 月に観測された各トラフィックにおける TCP 送信元ポート番号あたりのパケット数の割合を図 6.22，図 6.23，図 6.24，図 6.25 に示す．

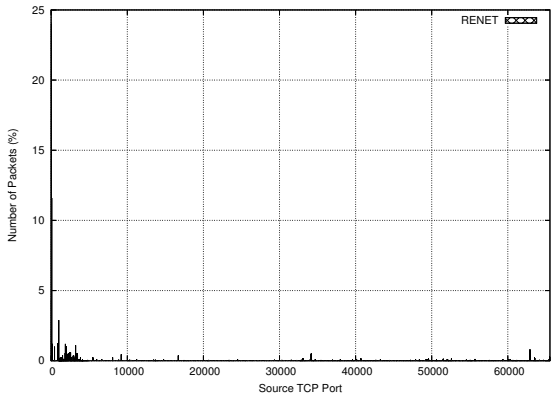

図 6.22: RENET における送信元 TCP ポート番号の分布

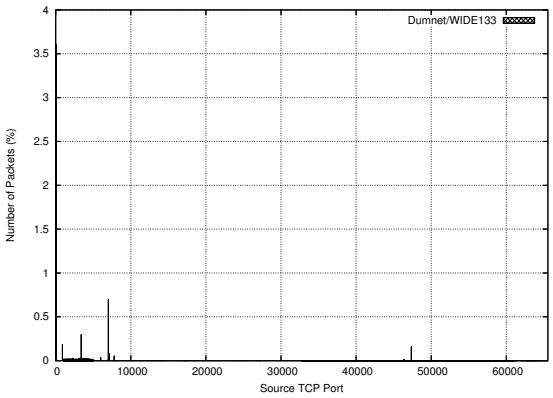

図 6.23: Dumnet/WIDE133 における送信元 TCP ポート番号の分布

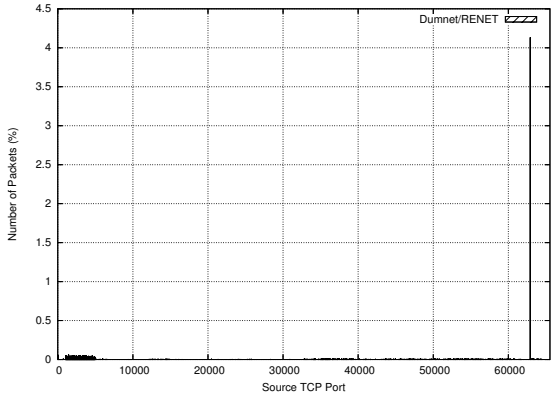

図 6.24: Dumnet/RENET における送信元 TCP ポート番号の分布

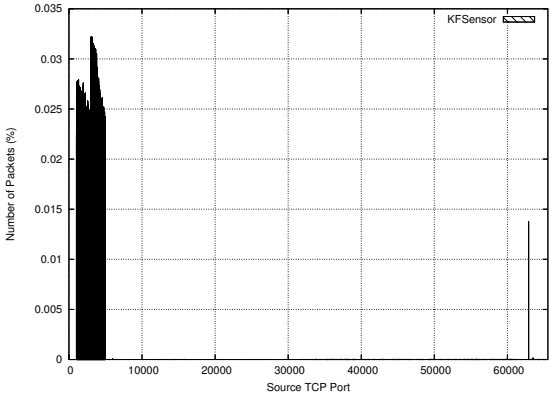

図 6.25: KFSensor における送信元 TCP ポート番号の分布

KFSensor を除き，いずれのトラフィックにおいても，送信元 TCP ポート番号は広く分散している．

6.2.9 TCP ヘッダ長の割合

TCP ヘッダは，送信元ポート番号，宛先ポート番号，TCP オプションなどを含む 20 バイト以上の可変長のヘッダフィールドである．

図 6.26 に各トラフィックにおける TCP ヘッダ長とパケット数の割合を示す．

TCP ヘッダ長は，TCP オプションを含まない場合には 20 バイト，何らかの TCP オプションが存在する毎に 4 バイト単位で増加する．いずれのトラフィックにおいても，75%以上のトラフィックにおいて，TCP ヘッダ長は 20 バイトであるが，研究・教育用

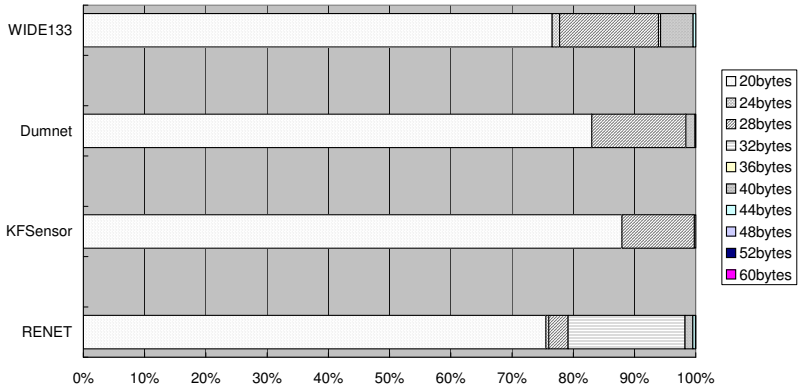

図 6.26: 各トラフィックにおけるヘッダ長とパケット数の割合

ネットワークのみにおいて，TCP ヘッダ長が 32 バイトのトラフィックが 19.09%を占めている．すなわち，TCP ヘッダ長が 32 バイトであるトラフィックは，ハニーポットの不正トラフィックである可能性はなく，研究・教育用ネットワークの通常トラフィックである可能性が高い．

6.2.10 TCP パケットのフラグの組み合わせとパケット数の割合

TCP のフラグは，TCP の状態を指定するフラグである．

図 6.27 に各トラフィックにおける TCP フラグの組み合わせとパケット数の割合を示す．

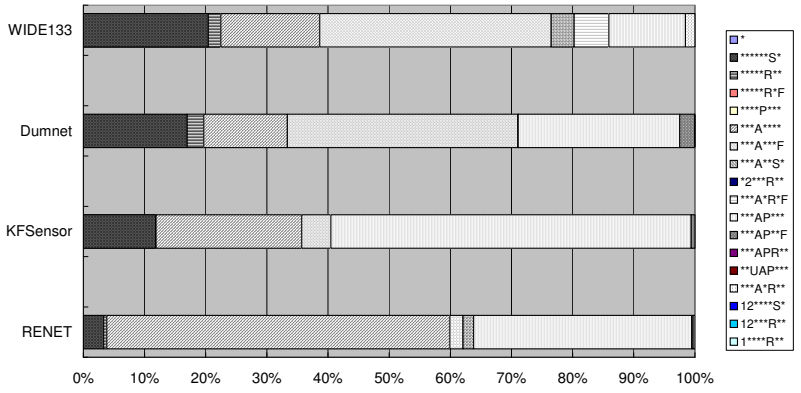

図 6.27: 各トラフィックにおける TCP フラグの組み合わせとパケット数の割合

各ハニーポットにおいては，ACK フラグのみが設定されたパケットは最大でも 23.82%であるのに対し，研究・教育用ネットワークにおいては 56.09%である．

TCP フラグは TCP フローの状態に応じて変化するため，特定の TCP フラグの組み合わせが設定されていることが直接，不正あるいは通常なトラフィックであることを意味しないが，どのような種類のトラフィックかを判断する材料の 1 つとして活用できる．

6.2.11 UDP 宛先ポート番号あたりのパケット数の割合

2005 年 12 月に観測された各トラフィックにおける UDP 宛先ポート番号あたりのパケット数の割合を図 6.28，図 6.29，図 6.30，図 6.31 に示す.

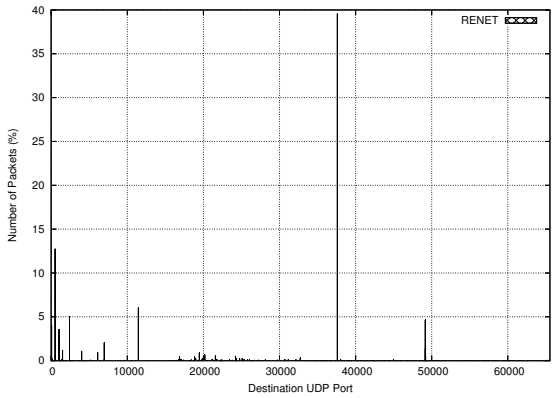

図 6.28: RENET における宛先 UDP ポート番号の分布

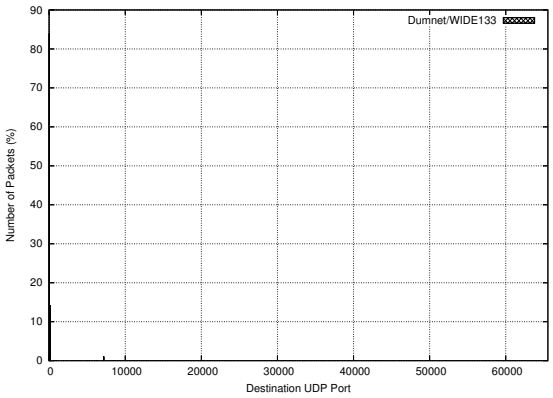

図 6.29: Dumnet/WIDE133 における宛先 UDP ポート番号の分布

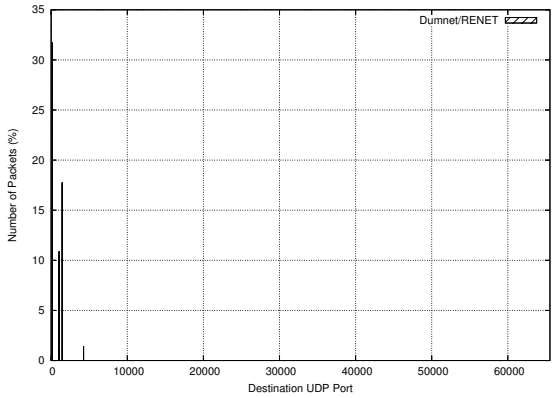

図 6.30: Dumnet/RENET における宛先 UDP ポート番号の分布

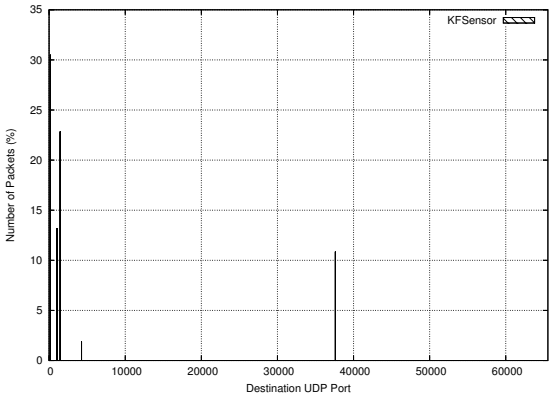

図 6.31: KFSensor における宛先 UDP ポート番号の分布

研究・教育用ネットワークにおいては，宛先 UDP ポート番号が広く分布しているが，各ハニーポットにおいては特定のポート番号に集中する傾向が見られる.

6.2.12 UDP 送信元ポート番号あたりのパケット数の割合

2005 年 12 月に観測された各トラフィックにおける UDP 送信元ポート番号あたりのパケット数の割合を図 6.32，図 6.33，図 6.34，図 6.35 に示す．

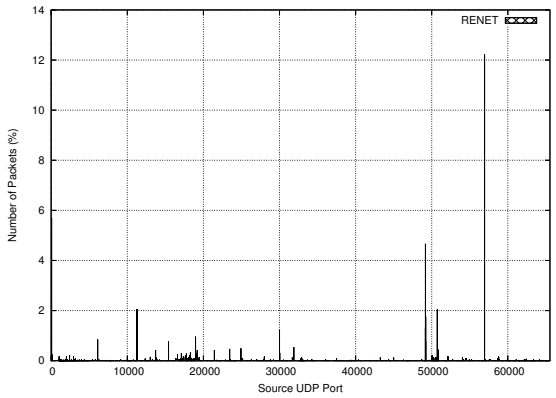

図 6.32: RENET における送信元 UDP ポート番号の分布

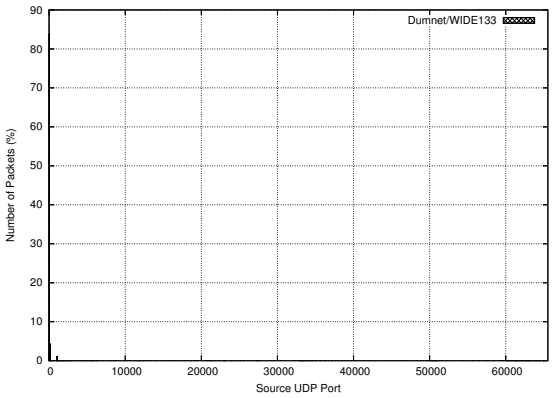

図 6.33: Dumnet/WIDE133 における送信元 UDP ポート番号の分布

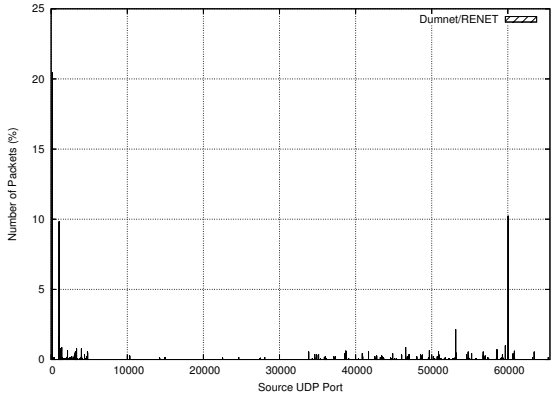

図 6.34: Dumnet/RENET における送信元 UDP ポート番号の分布

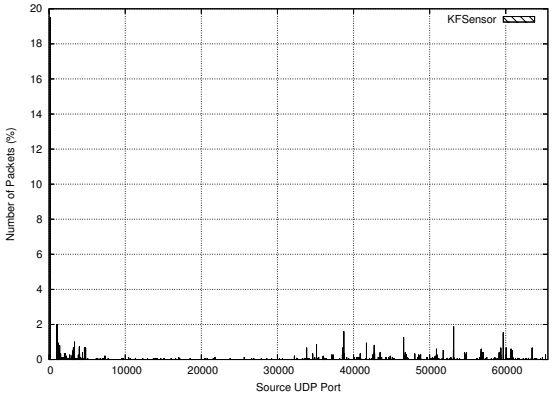

図 6.35: KFSensor における送信元 UDP ポート番号の分布

いずれのトラフィックにおいても，送信元 UDP ポート番号は広く分散している．

6.2.13 ICMP のタイプ番号の傾向

2005 年 12 月に観測された各トラフィックにおける ICMP タイプ番号あたりのパケット数分布を図 6.40 および図 6.36，図 6.37，図 6.38，図 6.39 に示す．また，図 6.40 に各トラフィックにおける ICMP タイプとパケット数の割合を示す。

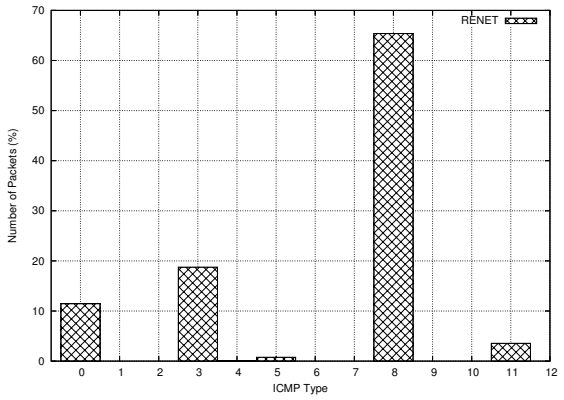

図 6.36: RENET における ICMP タイプの分布

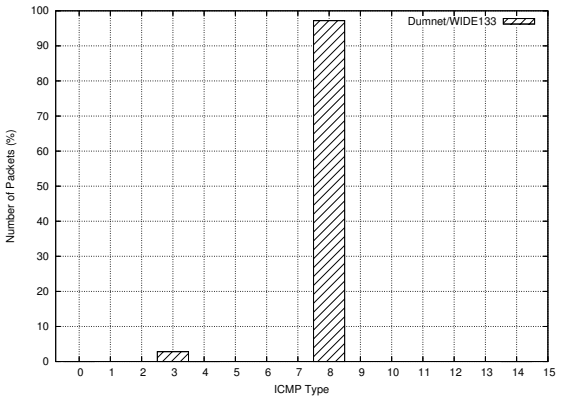

図 6.37: Dumnet/WIDE133 における ICMP タイプの分布

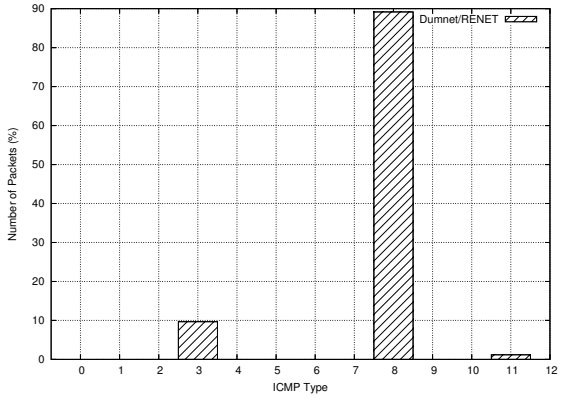

図 6.38: Dumnet/RENET における ICMP タイプの分布

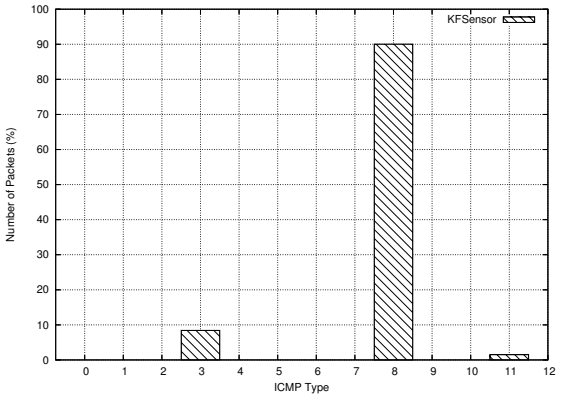

図 6.39: KFSensor における ICMP タイプの分布

6.2.14 ICMP のコード番号の傾向

2005 年 12 月に観測された各トラフィックにおける ICMP コード番号あたりのパケット数分布を図 6.45 および図 6.41，図 6.42，図 6.43，図 6.44 に示す．また，図 6.45 に各トラフィックにおける ICMP コードとパケット数の割合を示す。

研究・教育用ネットワークにおいては，ICMP Code 1 および Code 3 がハニーポットに比べて多く含まれている．

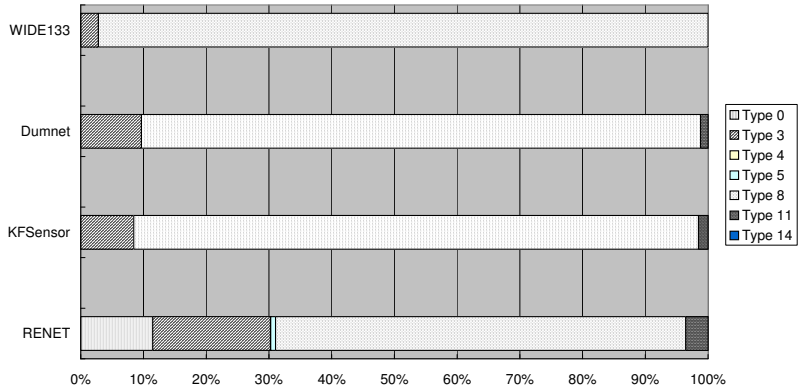

図 6.40: 各トラフィックにおける ICMP タイプとパケット数の割合

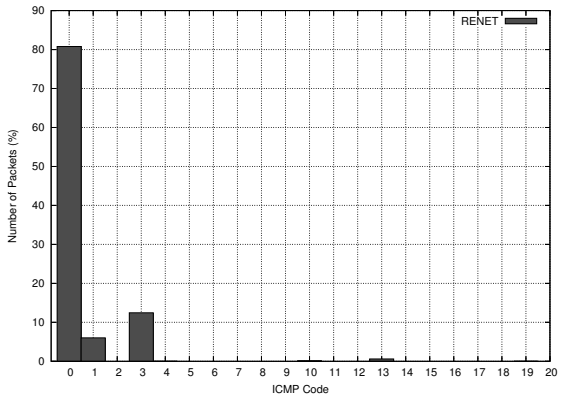

図 6.41: RENET における ICMP コードの分布

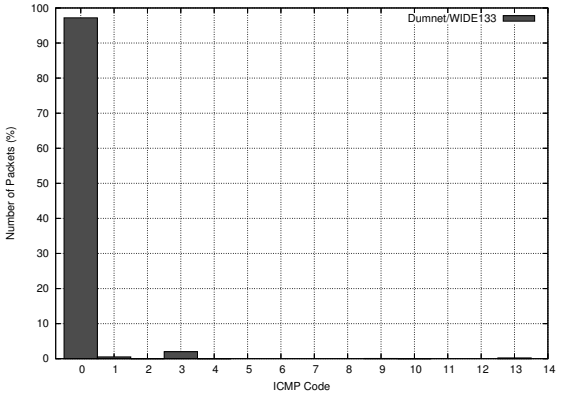

図 6.42: Dumnet/WIDE133 における ICMP コードの分布

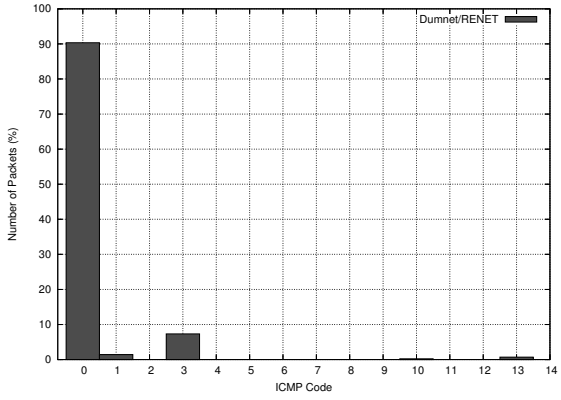

図 6.43: Dumnet/RENET における ICMP コードの分布

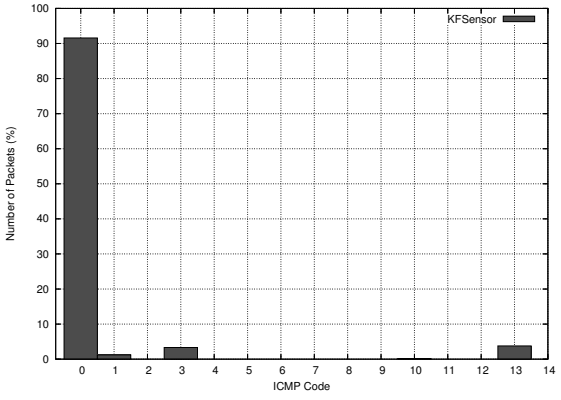

図 6.44: KFSensor における ICMP コードの分布

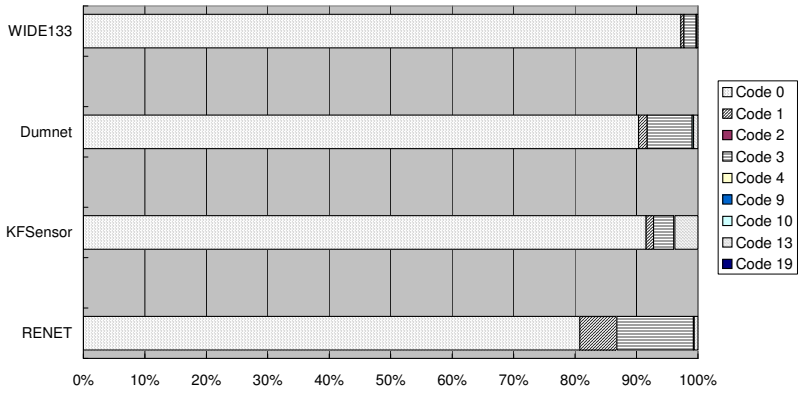

図 6.45: 各トラフィックにおける ICMP コードとパケット数の割合

6.3 まとめ

本章では，研究・教育用ネットワークおよび三種類のハニーポットのトラフィックにおけるネットワーク層とトランスポート層のヘッダフィールドの値の分布に違いが見られることを明らかにした.

第7章　パケットヘッダ情報に基づくスコアリングの実現

本章では，第4章で述べたトラフィック特性に基づいたパケットスコアリング手法のプロトタイプ実装について述べる．

7.1　トラフィックプロファイル

本手法では，ベイズ推測による事前確率を算出するために，事前にハニーポットおよび対照ネットワークで取得したトラフィックを元に，表7.1に示した各ヘッダと各パラメータ(各ヘッダに含まれる特定の値を含むパケット数)，および全パケット数をトラフィックプロファイルに記録する．

また，トラフィックプロファイルの例を表7.2に示す．

7.2　スコアリングアルゴリズム

このトラフィックプロファイルの内容を事前確率として用いたベイズ推測から，特定のヘッダフィールドの値を持つパケットがハニーポット宛のパケット，すなわち不正トラフィックのパケットである事後確率を算出する．

プロトタイプ実装ではベイズ推論のスコア算出アルゴリズムにはGary Robinson-Fisher法を採用する．Gary Robinson-Fisher法によるスコア算出アルゴリズムを以下に示す．

今，あるパケットに関して不正トラフィックかの事後確率を求める．このパケットはn個のヘッダフィールドを持つうち，x番目のヘッダフィールドの名をHdr_x，その値を$value_x$とする．

スコア算出に用いる各変数及び定数を以下のように定義される．

表7.1: 設定したヘッダフィールド

ネットワーク層	推定ホップ数(TTL)		ToS
共通ヘッダ	パケット長		フラグメンテーションビット
トランスポート層	TCP	宛先ポート番号	TCP フラグ(組み合わせ)
プロトコル別	UDP	宛先ポート番号	
ヘッダ	ICMP	ICMP タイプ	ICMP コード

表 7.2: トラフィックプロファイルの内容 (宛先ポート番号)

ヘッダフィールド	ハニーポットにおける パケット数	研究・教育用ネットワークにおける パケット数
21/tcp	37	1
22/tcp	87	108959
25/tcp	896	63069
80/tcp	7814	212030
110/tcp	0	361
…	…	…

Hdr_x 特定のヘッダフィールドの名

$BPkt$ ハニーポットで観測された全パケット数

$BIPs$ ハニーポットに割り当てられた IP アドレス数

$BParam(Hdr, value)$ ハニーポットで観測された全パケットのうち，ヘッダフィールド Hdr に値 $value$ を含むパケットの数

$APkt$ 対照ネットワークで観測された全パケット数

$AIPs$ 対照ネットワークに割り当てられた IP アドレス数

$AParam(Hdr, value)$ 対照ネットワークで観測された全パケットのうち，ヘッダフィールド Hdr に値 $value$ 含むパケットの数

$ROBS$ バックグラウンド上の情報の強さを表す定数

$ROBX$ 未知のヘッダが不正トラフィックであるかの推定確率

対照ネットワークのトラフィックには，正当トラフィックだけではなく，不正トラフィックも含まれる．このため，対象ネットワークとハニーポットの双方において，1IP アドレスに対して発生する攻撃トラフィック量は同一であるとの仮定に基づき，経験則により対照ネットワークの全トラフィックから不正トラフィック相当分を減じることで，“推測正当トラフィック” とする．

IP アドレス数による不正トラフィックの量の差異を補正するために，対象ネットワークの IP アドレス数 $AIPs$ をハニーポットの IP アドレス数 $BPkt$ を割った IP アドレス数比率を乗じた値 $GParam(Hdr, value)$ は，7.1 のように表される．

$$GParam(Hdr_x, value_x) = AParam(Hdr_x, value_x) - (BParam(Hdr_x, value_x) * \frac{AIPs}{BIPs}) \tag{7.1}$$

本プロトタイプ実装ではバックグラウンド情報の強さを示すRobinson法のパラメータ $ROBS$ は経験則により 0.0178 とする．また，ネットワークで観測されなかったヘッダの値が不正トラフィックであるかの推測確率 $ROBX$ は経験則により 0.52 とする．

これらの数値を用いて x 番目のヘッダフィールドの値に対する事後確率 $P(x)$ は式 7.2 のように表される．

$$P(x) = \frac{(\frac{BParam(Hdr_x, value_x)}{BPkt})}{(\frac{BParam(Hdr_x, value_x)}{BPkt} + \frac{GParam(Hdr_x, value_x)}{GPkt})} \tag{7.2}$$

ベイズの法則により，不正トラフィックと対照トラフィックにおける特定のパラメータの生起確率の和のうち，不正トラフィックにおける特定のヘッダの生起確率の割合を求めることにより，対象のパラメータの不正トラフィックらしさを求める．

それぞれ x 番目のヘッダフィールドの値に対するスコアを求める関数 $F(x)$ を式 7.3 に示す．関数 $F(x)$ の値域は $[0, 1]$ であり，1 に近ければ不正トラフィックの可能性が高いことを示し，0 に近ければ非不正トラフィックの可能性が高いことを示す．

ただし，$Paramcount(x) = GParam(Hdr_x, value_x) + BParam(Hdr_x, value_x)$ とする．

$$F(x) = \frac{((ROBS * ROBX) + (Paramcount(x) * P(x)))}{(ROBS + Paramcount(x))} \tag{7.3}$$

ここであるパケットに関して不正トラフィックかの事後確率 $Score$ は，式 7.4, 式 7.5, 式 7.6, 式 7.7 に示す S, T, S', T' を用いて式 7.8 のようにあらわされる．

$Score$ の値域は $[0, 1]$ であらわせ，1 に近ければ不正トラフィックの可能性が高いことを示し，0 に近ければ非不正トラフィックの可能性が高いことを示す．

$$S = 1 - \sqrt[n]{(1 - F(1)) \times (1 - F(2)) \times \ldots \times (1 - F(n))} \tag{7.4}$$

$$T = 1 - \sqrt[n]{F(1) \times F(2) \times \ldots \times F(n)} \tag{7.5}$$

$$S' = \chi^2(-2 \times \log(S), 2 \times ROBN) \tag{7.6}$$

$$T' = \chi^2(-2 \times \log(Q), 2 \times ROBN) \tag{7.7}$$

$$Score = \frac{(1 + S' - T')}{2} \tag{7.8}$$

(χ^2 は，カイ二乗検定を示す．)

7.3 実装環境

プロトタイプ実装では，Naïve Bayesian の一方式である Gary Robinson-Fisher[32] 法を用いた SPAM フィルタリングツールの bogofilter[33] を元に実装を行った．

実装に用いた環境を表 7.3 に示す．

プロトタイプ実装では，事前に保存したパケットに対してオフラインで分析を行う．

表 7.3: 実装ソフトウェア環境

OS	FreeBSD 5.4-RELEASE-p2
プログラミング言語	C 言語
コンパイラ	gcc 3.4.2
ベースソフトウェア	bogofilter-0.96.0

第8章　提案手法の評価

本章では，提案手法の評価について述べる．

8.1　定性評価

本節では，提案手法が第4章で述べた不正トラフィックの抑制に必要な要件に適合しているかどうか，また第2章で取り上げた他の機構との比較について述べる．

8.1.1　網羅性

新たな攻撃への対応

提案手法は，通常のトラフィックプロファイルとハニーポットのトラフィックプロファイルの内容に基づいてスコアリングを行う．従って，新たな攻撃に伴うトラフィック対して適切なスコアリングを行えるかどうかは，トラフィックプロファイルに依存する．

本研究が対象としているワームは，一般に無差別に攻撃を行うため，ハニーポットのIPアドレスに対しても攻撃が行われる．

従って，提案手法はハニーポットに対して攻撃が行われるという前提で，新たな攻撃に対応できるといえる．

フラグメント化されたトラフィックへの対応

提案手法は，データの正規化を行わずにパケットのスコアリングを行うため，フラグメント化されたパケットもそのまま処理される．このため，要件において指摘されたフラグメント攻撃の影響を受けない．

従って，提案手法はフラグメント化されたトラフィックに対応しているといえる．

TCPストリームへの対応

提案手法は，TCPストリームであってもIPパケット単位でスコアリングを行い，再構成処理を行わない．

これは，同一TCPストリームであっても，TCPコネクションの確立・維持・終了の各過程で用いられるパケット毎に異なるセキュリティリスクが存在しているからである．

例として，スリーウェイハンドシェイクを挙げる．まず，コネクションが確立する以前の段階においては，コネクションを開く側からSYN フラグを立てたパケットを送信する．この際，送信元IP アドレスを詐称していたとしても，宛先に対してパケットが到達することになる．また，この段階においては大量のSYN フラグの有効なパケットが送信されるSYN Flooding 攻撃が行われる可能性もある．

次に，SYN パケットを受信したコネクションを受動的に開く側では，TCP 接続を行う場合には，SYN フラグおよびACK フラグの有効なパケットを返答する．この段階において，SYN フラグ付のパケットの送信元IP アドレスが正当であれば正当なアドレスへ，詐称されたIP アドレスであれば詐称されたアドレスに応答が行われる．

最後の段階では，当初コネクションを開いた側がACK フラグの有効なパケットを返答することでTCP コネクションが確立する．このため，IP アドレスを詐称している場合には，TCP コネクションを確立することは困難である．

このように，TCP ストリームは，プロトコルの状態によってIP アドレス詐称などのセキュリティリスク要因が異なる．

提案手法では，TCP の宛先ポート番号およびフラグの組み合わせに対して生起確率を算出し，スコアを求めており，既存のパケット識別機構とは異なるアプローチでTCP ストリームのパケット識別に対応しているといえる．

新プロトコルへの対応

提案手法は，IP プロトコル共通のヘッダフィールドと，TCP, UDP, ICMP の各プロトコルのヘッダフィールドを用いてパケットのスコアリングを行うものであるため，新たなアプリケーション層プロトコルであっても，ネットワーク層にIP またはICMP，トランスポート層にTCP，UDP を用いている限り，提案手法でカバーできる．

従って，提案手法は新プロトコルに対応できるといえる．

暗号化トラフィックへの対応

一般的に暗号化されたトラフィックの内容を元に，パケットの識別を行うことは困難であるが，暗号化されたトラフィックであっても，ヘッダフィールド自体が暗号化されていなければ，提案手法を適用できる．

従って，IPsec やSSL，TLS といった広くインターネットで用いられているトラフィックに対しても提案手法を用いることが可能であり，暗号化トラフィックへ対応しているといえる．

8.1.2 スケーラビリティ

広帯域リンクへの対応

提案手法では，IP ヘッダの指定されたヘッダフィールドの値に基づいてスコアを算出するため，パケット識別にあたってはヘッダフィールドのみを参照し，スコアの算

表 8.1: 提案手法の概略

参照フィールド	TTL(推定ホップ数)，パケット長， トランスポート層プロトコル，宛先 TCP/UDP ポート番号 TCP フラグ，ICMP タイプ，ICMP コード
データの正規化	データの正規化は不要
フロー状態の保持	不要
必要な計算機資源	低い
不正トラフィックの識別	トラフィックプロファイルに依存

出を行う．

このため，IDS/IPS 等とは異なり，データの正規化などの処理が不要である．また，いかなる種類の IP トラフィックに対しても計算量が一定であるため，ハードウェア化が比較的容易で，広帯域リンクにも対応しやすいと考えられる．

トラフィックフロー

トラフィックフローの状態を保持する方式は，フローの数が多くなればなるほどメモリを消費し，多数のフローが存在するネットワーク環境に適用することが困難である．

提案方式は，トラフィックフローの状態を保持しない識別方式であるため，このような問題は生じない．

表 8.1 に提案手法の概略を示す．

8.2 定量評価

まず，学習に用いたネットワークのトラフィックを識別できているかを評価するため，学習データを取得した時間とは異なる時間を対象に，ハニーポットおよび研究・教育ネットワークのトラフィックのスコアリングを行った．

8.2.1 トラフィック収集期間

本研究では，スコアリングを行う対象のトラフィックを 2005 年 12 月 29 日 (木)0 時 00 分〜2005 年 12 月 29 日 (木)23 時 59 分に収集した．なお，特段の断りがない限り，時刻は全て日本時間 (JST) で表記した．

8.2.2 各サンプル毎のスコアの累積度数分布 (2005 年 12 月)

第 6 章で収集した各トラフィックプロファイルを事前確率として用い，研究・教育ネットワークのトラフィックのスコアの累積度数分布を図 8.1，図 8.2，図 8.3 に示す．なお，スコアは小数点以下 2 桁目で切り捨てている．

提案手法では不正トラフィックを 1.0，正当トラフィックを 0.0 とすることを目指しているため，できるだけ多くの割合の研究・教育ネットワークのトラフィックを低いスコアとするトラフィックプロファイルほど，優れたトラフィックプロファイルといえる．

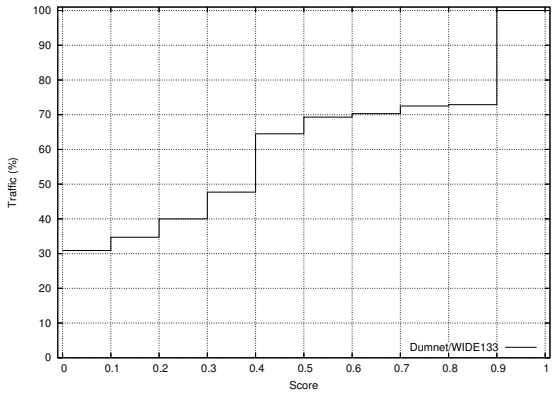

図 8.1: Dumnet/WIDE133 を用いた全トラフィックのスコア分布

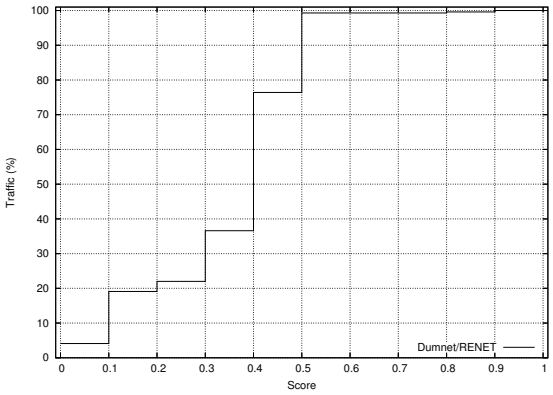

図 8.2: Dumnet/RENET を用いた全トラフィックのスコア分布

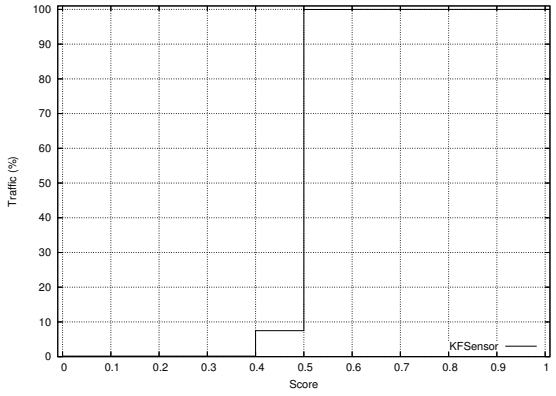

図 8.3: KFSensor を用いた全トラフィックのスコア分布

各プロファイルを用いた RENET に対するスコアの分布は，Dumnet/RENET99.3% のトラフィックがスコア 0.5 以下となった．また，KFSensor を用いた場合には，83.3%，Dumnet/WIDE133 を用いた場合には 69.3% となった．従って，通常トラフィックに対して低いスコアを付与する観点から，Dumnet/RENET のトラフィックプロファイルが適していることがわかった．

8.2.3 IDS 検出結果との類似性

本節では，IDS の検知結果との類似性を示すことにより，提案手法により既存の不正パケット識別手法と類似した結果を得ることができることを示す．

研究・教育ネットワークのトラフィックから，IDS により攻撃として検知されたトラフィックを分離し，得られたトラフィックのスコアの累積度数分布を図 8.4，図 8.5，図 8.6 に示す．

なお，IDS には不正検知方式の IDS である Snort 2.4.3 を利用した．また Snort のシグネチャには，2006 年 1 月 11 日 0 時に Subscriber ライセンスで取得した Sourcefire VRT certified Rules および BleedingSnort のルールセットを用いた．

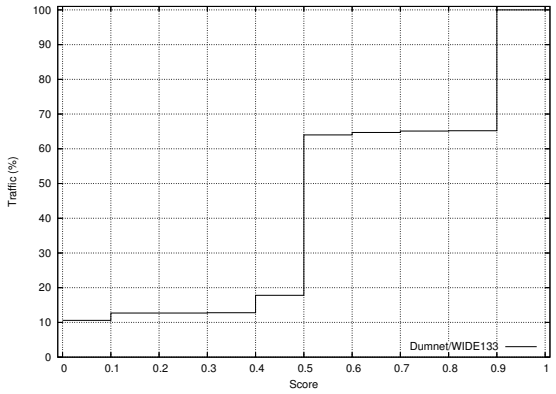

図 8.4: Dumnet/WIDE133 プロファイルによる IDS 検知結果のスコア分布

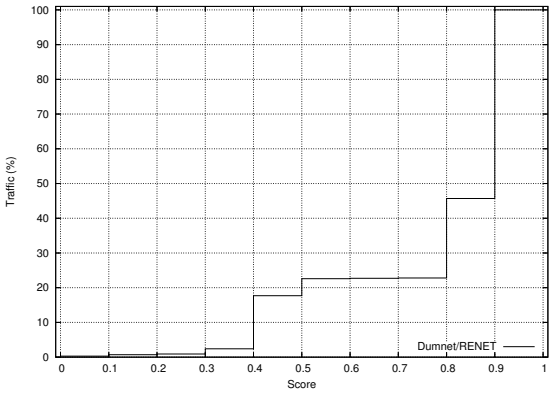

図 8.5: Dumnet/RENET プロファイルによる IDS 検知結果のスコア分布

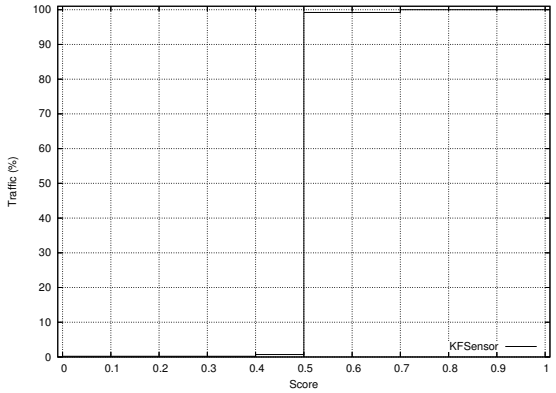

図 8.6: KFSensor プロファイルによる IDS 検知結果のスコア分布

IDS によって検知された不正トラフィックのスコア分布では，トラフィックプロファイルに Dumnet/RENET を用いた場合には，スコア 0.8 以上を示したトラフィックが 77.2% を占めた．一方，KFSensor/RENET を用いた場合には 0.0%，Dumnet/WIDE133 を用いた場合には，34.9% となった．

さらに，IDS の検知結果のうち，Slammer ワーム等の Microsoft SQL Server に対する不正トラフィックのスコアの累積度数分布を図 8.7，図 8.8，図 8.9 に示す．

Slammer ワームを初めとする Microsoft SQL サーバを狙った攻撃に伴うトラフィック

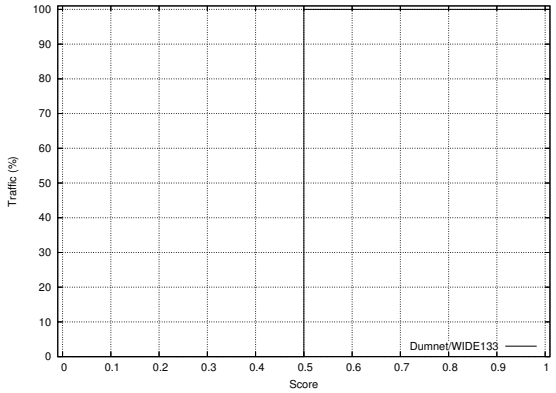

図 8.7: Dumnet/WIDE133 プロファイルによる検知結果のスコア分布

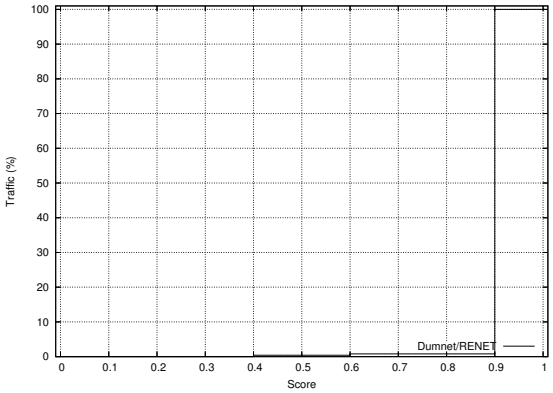

図 8.8: Dumnet/RENET プロファイルによる検知結果のスコア分布

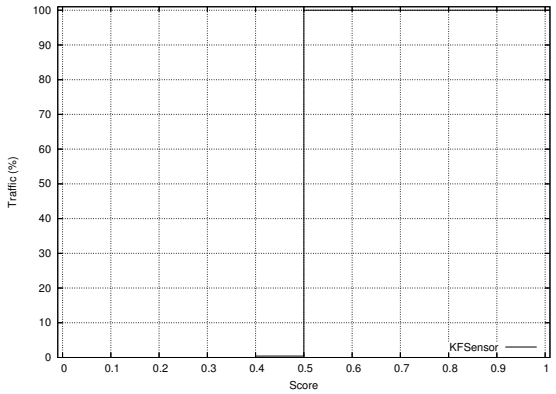

図 8.9: KFSensor プロファイルによる検知結果のスコア分布

のスコアリングを行った結果，Dumnet/RENET プロファイルを用いた場合にスコア 0.8 以上となるトラフィックの割合は 99.2% であったのに対し，KFSensor プロファイルおよび Dumnet/WIDE133 プロファイルでは 0.0% であった．

これらの結果から，提案手法においてはトランスポート層型ハニーポットのトラフィックプロファイル，特に同一 IP アドレス空間のトラフィックプロファイルを用いた場合に，IDS に近似した出力を得ることができることが示された．

8.2.4 誤検知 (False Positive)

提案手法では，正当なパケットに対して高いスコアを付与する可能性がある．このため，評価トラフィックにおいて高いスコアを得たパケットから，100 パケットをサンプリングし，ネットワーク管理者が次の分類基準に基づいてトラフィックの性質を分類する．

Yes 明らかな不正トラフィック (ワーム, 送信元 IP アドレス詐称に対する応答パケット, ホストプローブ)

Policy ポリシによっては遮断する場合のあるトラフィック (ICMP ECHO 要求など)

Unsure 性質が不明なトラフィック

No 明らかな正当トラフィック

表 8.2 に示したとおり，実験においては明らかに正当なサービス利用に伴うトラフィックに対して 0.8 以上のスコアを付与する確率は 2%であり，不明なトラフィックを含めた場合でも 5%にとどまる．

表 8.2: 高スコアパケットの分類結果

スコア分類	0.80〜0.89	0.90〜0.97	0.98〜0.99	1.0	0.8〜1.0 計
Yes	17%	4%	63%	57%	27%
Policy	81%	94%	24%	19%	68%
Unsure	0%	0%	11%	24%	3%
No	2%	2%	2%	0%	2%
パケット数	12083	6411	7243	43	25780

8.3 評価のまとめ

本章では，定性評価において，提案手法が第 4 章で述べた不正トラフィックの抑制に必要な要件に適合していることを確認し，他のパケット識別機構と比較した場合の優位性を明らかにした．

また，定量評価において，既存手法である IDS が Slammer ワームとして検知したトラフィックに対するスコアリング結果を示し，提案手法が IDS と類似した出力を行うことを示した．

第9章　結論

9.1　まとめ

本研究では，低対話型ハニーポットのトラフィックと，研究・教育ネットワークの実トラフィックの IP ヘッダの傾向が異なることを明らかにした．

さらに，トランスポート層プロトコル，パケット長，宛先ポート番号，推定ホップ数などの IP ヘッダの内容を元にベイズ推測を行うことで，IP パケットに対してスコアを算出し，ネットワーク管理者の定義に基づきクラス分けを行う手法を提案した．また，評価の結果，Slammer ワームの 85.59%のトラフィックが，不正トラフィックの閾値である 0.98 点以上のスコアとなった．これにより，提案手法は既存の IDS で検知した不正トラフィックを高い割合で識別できることを明らかにした．

9.2　今後の課題

プロトタイプ実装では，パラメータをトランスポート層プロトコル，パケット長，宛先ポート番号，推定ホップ数などに限定した．今後，スコアに対して各パラメータの寄与度を評価するとともに，スコアリングの精度を向上させるために，新たなパラメータの導入が課題となる．また，パケットのスコアリングに Gary Robinson-Fisher 方式を用いたが，不正トラフィックの識別に適した手法およびパラメータの検討が必要である．

さらに，本手法は IP パケットのクラス分けのみを行っているため，不正なトラフィックの拡散を防止できない．そのため，今後は本手法に適したトラフィックの制御手法の研究が必要である．

本手法は，IP ヘッダの各フィールドの構成比率が，ハニーポットと研究ネットワークのトラフィックが異なることに依存している．従って，トラフィックの構成に差異が少ない場合や，ヘッダの値が改ざんされた場合には，スコアリングの精度は低下する可能性が想定されるため，さらなる研究が求められる．

本研究では，パケットに対するスコアリングのみを行っているが，今後は，スコアリングデータを活用したパケットの制御手法の研究が検討される．

謝辞

本論文の作成にあたり，御指導いただきました慶應義塾大学環境情報学部教授村井純博士，奈良先端大学院大学の山口英博士，並びに慶應義塾大学環境情報学部助教授中村修博士，同学部助教授楠本博之博士，政策・メディア研究科特別専任講師南政樹氏に感謝致します.

絶えず御指導と御助言を頂きました慶應義塾大学環境情報学部専任講師の重近範行博士，シスコシステムズ株式会社の森川誠一氏，慶應義塾大学院政策・メディア研究科博士課程の小原泰弘氏，株式会社ラックの村上純一氏に深く感謝致します.

本論文の作成にあたり，御協力いただいた慶應義塾大学院政策・メディア研究科修士課程の佐川昭宏氏，慶應義塾大学環境情報学部の水谷正慶氏，金井瑛氏，並びに慶應義塾大学徳田・村井・楠本・中村・湧川研究室の皆様，特にSING研究グループの皆様に感謝の念を表します.

参考文献

[1] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. The Spread of the SapphireslashSlammer Worm. Technical report, January 2003.

[2] Cable News Network LP. Ebay, buy.com, amazon, cnn.com sites hit, Feb 2000. http://money.cnn.comslash2000slash02slash08slashtechnologyslashyahooslash.

[3] J. Wroclawski. The Use of RSVP with IETF Integrated Services. RFC 2210 (Proposed Standard), September 1997.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), December 1998. Updated by RFC 3260.

[5] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (Proposed Standard), September 1997. Updated by RFCs 2750, 3936.

[6] Thomas H. Ptacek, Timothy N. Newsham. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. Jan 1998.

[7] Inc. Cisco Systems˙Network-based application recognition (NBAR).

[8] 株式会社 CRC 総合研究所, SRI Informational, and CRC テクニカ ル株式会社. ベイズ推論によるネットワーク侵入検知システムの開発. **創造的ソフトウェア育成事業**, 1996.

[9] Yoohwan Kim, Wing Cheong Lau, and Mooi Choo Chuah H. Jonathan Chao. Packetscore: Statistics-based overload control against distributed denial-of-service attacks. In *Proceedings of the 2004 IEEE Infocom Conference (INFOCOM)*, March 2004.

[10] Masaki Ishiguro, Hironobu Suzuki, Ichiro Murase, and Hiroyuki Ohno. Internet threat detection system using bayesian estimation. *FIRST Conference 2004*, 2004.

[11] Denis Zuev and Andrew W. Moore. Traffic classification using a statistical approach. March 2005.

[12] T. Dunnigan and G. Ostrouchov. Flow characterization for intrusion detection. November 2000.

[13] Sebastian Zander and Thuy Nguyen and Grenville Armitage. Self-learning ip traffic classification based on statistical flow characteristics. March 2005.

[14] Matthew V. Mahoney and Philip K. Chan. PHAD: Packet header anomaly. detection for identifying hostile network traffic, 2001. Florida. Tech. Technical Report 2001-04.

[15] Matthew V. Mahoney and Philip K. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *KDD*, pages 376–385, Jul 2002.

[16] Cisco guard DDoS Mitigation Appliances (Cisco Guard XT 5650). http://www.cisco.comslashenslashUSslashproductsslashps5888slash.

[17] Symantec internet security threat report. http://www.symantec.comslashregionslashjpslashistrslash.

[18] G. Ziemba, D. Reed, and P. Traina. Security Considerations for IP Fragment Filtering. RFC 1858 (Informational), October 1995.

[19] I. Miller. Protection Against a Variant of the Tiny Fragment Attack (RFC 1858). RFC 3128 (Informational), June 2001.

[20] Lance Spitzner . Honeypots: Definitions and Value of Honeypots , May 2003. http://www.spitzner.net/honeypots.html.

[21] Junichi Murakami. Dumnet. http://tf.rootkit.jp/work/dumnet/.

[22] Kfsensor - windows honeypot ids. http://www.keyfocus.netslashkfsensorslash.

[23] Niels Provos. A virtual honeypot framework. In *Proceedings of the 12th USENIX Security Symposium*, pages 1–14, August 2004.

[24] NETSEC. Specter 8.0, 2005.

[25] Symantec Corporation. Symantec Decoy Server, May 2005.

[26] Lance Spitzner. The Honeynet Project: Trapping the hackers. 1(2):15–23, March/April 2003.

[27] V. Yegneswaran, P. Barford, and D. Plonka. On the design and use of internet sinks for network abuse monitoring. In *Proceedings of Recent Advances in Intrusion Detection*, 2004.

[28] Shin Shirahata. 未使用のアドレス空間を用いたハニーポットの構築 (interdomain routing security workshop 5), July 2005.

[29] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349.

[30] WIDE Project. 第 4 部 DNS の拡張および運用環境. Mar 2002.

[31] Luca Deri, Gaia Maselli, and Stefano Suin. Design and implementation of an anomaly detection system: an empirical approach. 2003.

[32] Paul Graham. "better bayesian filtering ", Jan 2003. http://www.paulgraham.com/better.html.

[33] Bogofilter. http://bogofilter.sourceforge.netslash.

Copyright ©2005-2006 Shin Shirahata. All rights reserved.