組織内ネットワークにおけるネットワーク運用管理技法に関する研究

大阪府立大学博士論文

組織内ネットワークにおけるネットワーク運用管理技法に関する研究

2013 年 2 月

川橋　裕

目次

図目次

表目次

第 1 章

序論

インターネットおよびイントラネットを構成するネットワーク技術は，階層モデルを採用している．端末間や端末とサーバ間の通信で，通信内容を指す情報は，物理層からアプリケーション層までを，フレーム，パケットおよびデータなどの形式で相互に交換される．したがって，ユーザ利用およびアプリケーション開発では，各層の仕組みや状況を把握，理解する必要がない．しかし，ユーザが理解する障害は「つながらない」と体感することから始まる．そのため，ネットワーク運用管理には，各層における問題の切り分けからユーザの設定ミス，アプリケーションの挙動確認など，様々な面での技能を必要とする．1990 年代半ばより，全世界でインターネットを利用した教育，産業，ビジネスおよび研究が著しく発展してきた．ネットワークにおいては，メトロネットワークや拠点間接続，トンネル技術および経路制御などが急激に進歩し，複雑化している．コンピュータシステムにおいても，クラウドなどネットワーク技術を前提とした仕組みが同じく複雑化している．近年はサービスの分野で，セキュリティ，プライバシ，フォレンジックおよびポリシのように，ネットワーク運用管理全体が複雑なしがらみによって構成されるようになった．本来，ネットワーク運用管理の困難さは，一方でユーザの利便性を確保しつつ，他方で機密性，耐障害性（事前防御と事後対応）などの完全性を高めるための程度問題に帰結する．一般的に，ビジネスや行政分野では，コンプライアンスなどのため制限事項を増やし，業務を遂行する上での利便性を低減させている．しかし，学術研究機関などでは研究教育業務に支障を来すため，制限事項に頼らず運用管理を支援するシステムと，支援する体制が必要となる．上記の支援システムに関する例として，近年では事前防御が困難で，事後対応でもリスクを回避し難い問題が 2 つ顕在化している．1 つ目は P2P(Peer to Peer) ファイル共有ソフトウェア利用による著作権侵害であり，2 つ目は DoS(Denial of Service:サービス妨害）攻撃である．前者は政府機関や民間機関によって，交換するファ

イルの多くが違法コンテンツであると判明している．後者は攻撃対象のリソース枯渇を狙う手法である．これらは正規の手順に沿っているため検知と対応が困難とされている．一方，上記の支援体制に関する例として，運用管理に従事する管理者の不足が挙げられる．運用管理には，知識や技術に偏らず，これらを技能として総合的に習得，発揮する人材が必要である．しかし，総合的な技能は運用管理の経験によって習得できるため，知識や技術を積み重ねた上で，より現実に近いシミュレーションや演習を実現する支援システムと運用体制が必要である．

本研究は，上記の運用管理支援システムと運用管理人材育成支援システムの 2 部より構成される．第 2 章と第 3 章では，新たな運用管理支援システムとして，それぞれファイル共有ソフトウェア利用検知システムと，DoS 攻撃に対する透過型防御システムについて述べる．さらに第 4 章では，運用管理人材育成支援システムとして，実際の運用管理を演習形式で実施する環境と運用法について述べる．これらは筆者らが全国の学術機関に対して実施してきた情報危機管理コンテストや，文部科学省の先導的 IT スペシャリスト育成推進事業の一環で始めた IT 危機管理演習で活用し，現在も継続している．以下に，各章における要旨を述べる．

第 2 章では，トラヒックパターンに基づくファイル共有ソフトウェア利用検出システムについて述べる．ファイル共有ソフトウェア利用を検出する手法には，一般的にクローリング型とゲートウェイ型，およびパケットキャプチャ型がある．クローリング型は，インターネット上で P2P ネットワークがどのように構築されているかを調査する手法である．ゲートウェイ型は，各ファイル共有ソフトウェアごとに検出用のシグネチャを用意し，パケットのペイロードを解析して検出する手法である．これらは，調査に大規模なシステム導入が必要な点や，シグネチャを継続的に解析し提供する必要性，およびスループット低下の懸念などの問題点がある．したがって，研究分野ではパケットモニタ型を採用している事例が多い．ファイル共有ソフトウェアの多くは，膨大な数の通信相手先 (ピア) を必要とするか，初期のピアやファイルの所在をインデックス化して固定する形態に大別される．既存研究では，Winny や Share など挙動が顕著なファイル共有ソフトウェアに限定して検出するとともに，Skype など有用なファイル共有ソフトウェアを誤検知する場合が多い．

本研究では，ピア数とインデックス先の特定に加えて，TCP 送信ポート番号の連続性や UDP 送信ポート番号の同一性などを指標として採用する．同一の端末内でさまざまなアプリケーションが稼働する場合，OS（Operating System）上で規定された短命ポート

（Ephemeral Port）により送信元ポートが設定，利用される．このため，同一端末内での複数アプリケーション利用を区別できる．これは，冒頭に示した階層構造の中で，特定階層の挙動として規定されている．アプリケーション層とネットワーク層のみでファイル共有ソフトウェアの挙動を捉えるだけでなく，他の層を含めてトラヒックをパターン化することが，誤検知を回避するだけでなく，さまざまなファイル共有ソフトウェア利用の検知に役立っている．本論文では，提案手法をソフトウェア実装し，和歌山大学内ネットワークでの検出実験を行う．その結果，代表的な 15 種類の P2P ソフトウェアすべてに対し，起動時検知が確認され，有用な P2P ソフトウェアである Skype の誤検知を回避することを実証する．

続いて第 3 章では，ソースアドレスルーティングによる DoS 攻撃等への防御システムについて述べる．DoS 攻撃によるサービス妨害は，近年業務妨害だけでなく，脅迫や詐欺などの事件として多発していることが報告されている．これは DoS 攻撃の検知と回避が困難であることに起因する．既存研究では，DoS 攻撃を受けるサーバでの検知手法や，ACL(Access Control List) による回避策が検討されてきた．しかし，外部に公開するサーバは分散される傾向にあり，一般アクセスを保持しつつ ACL 機能を十全に提供するには，相当なハードウェア要件が求められる．

本研究では，サーバごとに DoS 攻撃と認定する閾値を変更し，ACL ではなく送信元 IP アドレスを用いた経路制御によって，対外接続部で統括して防御するシステムを提案する．冒頭に示した階層構造では，上位になるほど OS 内部でカーネルメモリのコピーが多発するため，一般的に処理の負荷が高くなる．これに対し，経路制御は ACL 機能より下位で実装されているため，DoS 攻撃への耐性が高いといえる．一方で，本来経路制御は宛先 IP アドレスに対して実施されるため，攻撃対象のサーバを指定しては，一般アクセスも遮断することになる．提案システムでは，送信元 IP アドレスに対して経路制御を施し，DoS 攻撃と認定されたアクセスのみを NULL デバイスや他の IP アドレスに転送することで，負荷を低減しつつ一般アクセスを保護している．上記は，ファイアウォールなどによるアクセス制御を基にした製品群や既存研究とは違い，下位層であるネットワークが本来得意とする経路制御を用いる点が特徴である．加えて提案システムでは，転送先に代理応答サーバを指定することで，DoS 攻撃の一種である F5 攻撃 (リロード攻撃) に対して攻撃認定されたことを通知する．これは提案システムで設定した閾値による false-positive を改善する機能である．代理応答サーバは TCP パケットによるリクエストをステートレスに応答し，さらに返信内容を簡素化することで負荷を低減している．さらに，本提案システムはブリッジ型で構築しているため，既存のネットワーク構成を変更することなく導入が可能である．ルータやスイッチ，あるいはハブなどネットワーク機器

間に挟み込む一方で，本提案システムに障害が発生した場合，ケーブルの挿抜のみで速やかに原状回復できる．本論文では，本提案システムの動作を実験環境下で検証する．和歌山大学の日中平均である 6,000pps(Packet Per Second) の状況下で，50,000pps の DoS 攻撃 (SYN Flood) を発生させた場合の一般アクセス (HTTP,DNS) の保護，および攻撃対象サーバと本提案システムの負荷について，良好な結果を示す．さらに，代理応答サーバの返信能力として，DoS 攻撃 (リロード攻撃) に対して，5,000pps まで完全に返信することを明らかにする．これはシステム設計により，およそ 100 台からのリロード攻撃に対して，毎秒すべての送信元に返信できることを指す．当該機能は false-positive の把握と改善に対して有効であり，攻撃認定の閾値周辺にある一般アクセスへの対応に十分な能力であるといえる．

第 4 章では，運用管理人材育成支援システムと同システムの運用法について述べる．2010 年以降，日本では日本版 CTF(Capture The Flag) として，互いに侵入と防御を競うイベントが増えつつある．これは米国の DEFCON など，ハッキング (Hacking) が，本来プログラミングに造詣が深いという意味に則って企画されたイベントと同様の効果を狙っているためである．しかし，ネットワーク運用管理では，すべての障害が攻撃に起因するとは限らない．故障，ユーザの設定や使い方のミス，複合的な要因で単一の障害が発生する場合や，逆の場合もある．運用管理に必要な技能には，原因の特定に必要な切り分け手法によって上記を判定することや，原状回復を確認できること，およびユーザとの適切な情報交換など多様な能力が求められる．筆者らは，2006 年より情報危機管理コンテストと称して，全国の学術機関より参加チームを募って，上記技能の必要性を体感できる環境システムを構築，運用してきた．

当該コンテストは，2009 年より最優秀チームに経済産業大臣賞が授与されている．さらに，本提案システムは，2007 年より文部科学省の「先導的 IT スペシャリスト育成推進事業」における「IT-KEYS : 社会的 IT リスク軽減のための情報セキュリティ技術者・管理者育成」の IT 危機管理演習に採用され，現在も当該演習実施を継続している．

本提案システムでは，シナリオ方式を採用しており，提案システムを運用する運営側と，参加者や受講者など障害に対応する参加側が存在する．運営側は，攻撃を含むさまざまな要因で障害が発生するよう事前に設計，検証した環境下で，攻撃や苦情連絡を実施する．参加側は，苦情への対応と原因の特定，修復だけでなく，適切な情報開示を実施することでリスクを軽減させる．参加側の苦情対応では，苦情連絡する運営側の均質化を図ることで公平性を高める．すなわち，苦情の内容だけでなく，参加側から質問される内容を想定した問答集を含めて，シナリオを構築する．同シナリオでは，攻撃手順やシナリオの終了条件を明確にし，暫定措置の場合には再度攻撃することで障害を再発させる．運用管

理の人材育成システムは，教育支援システムと近似する．シナリオ方式によって演習の流れを制御し，対応できない，対応が間違った参加側に対して，適切な対応修正を施す．CTF など参加側の技術力に依存するのではなく，運用管理に必要な技能を体感できることが本システムの特徴である．一方で，開催ごとにシナリオを新規構築するには負担がかかる．さらに，実施には大規模な設備が必要であり，参加側に会場まで来場させるには予算も必要となる．本提案システムでは，2009 年より ASP(Application Service Provider) 形式を採用し，コンテスト予選と演習において導入，実施している．

本提案システムにおいて，コンテスト参加者や演習受講者の技量の向上を定量的に評価することは困難である．しかし，他の類似のイベントと比較した定性評価と，コンテストや演習の実施を継続している実績から，他に類を見ない有効な運用管理人材育成システムであることを検証する．

第 5 章では，本研究で得られた結果を総括するとともに，今後の取り組むべき課題について整理する．

第 2 章

ファイル共有ソフトウェア利用検出システムの精度向上と運用支援法

2.1 序言

高品質なネットワーク接続環境が広まるにつれて，音楽や映像などのファイルをネットワークを通じて共有するサービスが普及している．このサービスの 1 つにファイル共有ソフトウェアがある．P2P(Peer-to-Peer) 方式 [1][2] の通信は，従来のクライアント・サーバ方式の通信に比べて，匿名性，耐障害性，スケーラビリティの高さなどのメリットを有する．ファイル共有ソフトウェアは，P2P 方式の通信を利用することで，不特定多数とファイルを共有できるソフトウェアである．

しかし，ファイル共有ソフトウェアの利用で様々な問題が発生している．最初に，共有されている音楽や映像などのコンテンツの大半が，著作者に無断で利用されているという問題である [3]．次に，ファイル共有ソフトウェアはファイルの検索やダウンロードを高速にするために，通信帯域を占有するという問題である．最後に，ファイル共有ソフトウェアを利用することによるウイルス感染や，情報流出の問題である．

文部科学省のガイドライン [4] や IPA（独立行政法人情報処理推進機構）の方針 [5] にしたがい，多くの大学機関ではセキュリティポリシにおいてファイル共有ソフトウェアの利用を禁止している．その上で，一部では同ソフトウェアの利用を，申請による許可制で運用している．しかし，ファイアウォールの設計における「全面禁止からの部分解除」を実現するには，悪意の定義と潜在的な利用の制限が困難であること，および Skype[8] などの false-positive 回避が困難であることなどの問題に見られるように，システム設計および運用管理設計が有機的につながっていない．

ファイル共有ソフトウェアの利用を検出する製品として，One Point Wall[6] や，Palo Alto 社製の統合ファイアウォールである Palo Alto[7] などがある．これらはパケットのペイロードを解析し，ソフトウェア名やバージョン番号，あるいはこれらを含む暗号化されたデータの記述パターンなど，ファイル共有ソフトウェア特有の表記の有無によって検出を試みる．しかし，パケットのペイロードを解析する手法は，ネットワーク利用者の通信内容のプライバシを脅かすと理解されることもあり，情報倫理の観点から望ましくない．さらに，検知対象のソフトウェアを増やすには高額のライセンス料や保守料を必要とし，幅広い機関での導入と運用を困難にしている．加えて，DPI（Deep Packet Inspection）技術で 10 種類以上あるファイル共有ソフトウェアに対応させるには，開発と解析の継続に多大なコストが必要となり，ベンダの新規参入や利用検知の研究を困難にしている．

本論文では，ファイル共有ソフトウェア利用者の端末の通信状況を指すトラヒックパターンを収集することで，ファイル共有ソフトウェアの利用を検出する手法を提案する．トラヒックパターンについては 2.3.1 節で厳密に定義する．本提案では，特にファイル共有ソフトウェアの起動時検出を目標とする．パケットのペイロードを解析することなく，トラヒックパターンを利用することで，通信のプライバシに配慮する．さらに，Skype などの有用な P2P ソフトウェアに対して誤検出しない手法を提案する．最後に，提案手法を実装したシステムによる和歌山大学内ネットワークでの実験結果を示し，本提案手法の有効性を明らかにする．

2.2 P2P 技術

2.2.1 P2P

複数台の端末による通信形態として，クライアント・サーバ方式と P2P 方式が存在する．

P2P 方式の通信では，クライアント，サーバの役割を明確にもたずに，端末はクライアントとしてもサーバとしても動作する．P2P 方式のメリットとして，スケーラビリティの高さ，耐障害性，匿名性などが挙げられる [1][2]．これらは，以下に示す 3 種の P2P ネットワークに大別される．

ピュア型 P2P　各端末が隣接した端末に通信することで，データを検索する．端末の欠落が発生しても，ネットワークを維持するため，耐障害性が高い．ピュア型 P2P

による代表的なファイル共有ソフトウェアに Winny がある．

ハイブリッド型 P2P　ハイブリッド型 P2P では，データのインデックス情報をインデックスサーバが管理し，データは各端末が管理する．端末がデータを新たに作成すると，インデックスサーバにデータの場所をインデックスとして登録する．インデックスサーバがデータの場所を集中管理するために，データの検索効率が良い．ハイブリッド型 P2P を用いた代表的なファイル共有ソフトウェアに WinMX がある．

スーパーノード型 P2P　スーパーノード型 P2P では，インデックス情報をスーパーノードクラスタと呼ばれる端末群が管理する．スーパーノードクラスタは処理能力の高い端末から選出される．スーパーノード型 P2P はピュア型 P2P とハイブリッド型 P2P の利点を併せ持つ．スーパーノード型 P2P を用いた代表的なソフトウェアに Skype がある．

上記の各特徴から，ハイブリッド型とスーパーノード型は通信相手先数が少ないと推測できるが，後述する調査実験において，ファイル共有ソフトウェアの起動時における特徴抽出の観点からは，上記の分類よりも初期ノードの取り扱いによる分類の方が適切であることがわかった．以下に新たな分類について述べる．

2.2.2　起動時の挙動による分類

第 1 章で述べたように，本研究ではファイル共有ソフトウェアの起動時検出を目指す．ファイル共有ソフトウェア起動時の挙動の分類は，2.2.1 節の型を踏襲していない．主として P2P ネットワークを最初に構成する初期ノード登録の要・不要で大別されるが，詳細は以下のように分類される．

手動登録型　ピュア型 P2P に多く見られるのが手動登録型である．最初に P2P ネットワークを構成する通信相手先（ピア）である初期ノードを，ユーザが手動で登録する．登録できる初期ノード数には，ファイル共有ソフトウェアによって 1 個のみ，もしくは 1 個〜多数個の場合がある．ファイル共有ソフトウェア起動後，ピアとなるノードが自動的に追加され続ける．一般には，多くの通信相手を有するノードが追加されると，通信相手先数および検索効率が飛躍的に上がる．

自動登録型　ピュア型 P2P にいくつか見られるのが自動登録型である．ファイル共有ソフトウェアが，同ソフトウェアの開発サイトなどから初期ノードを自動で取得する．以後の挙動については手動登録型と同じく，ピアとなるノードが自動的に追加され続ける．開発サイトが閉鎖されると利用不能になることがある．

インデックス型　ハイブリッド型やスーパーノード型に見られるのがインデックス型である．初期ノードはインデックスサーバにある．同サーバ上に，取得するファイルを保持するノードが登録されているため，あらかじめ高い検索効率を有する．ピアとなるノード数は検索効率に関係ないため，通信相手先は多くない．

ファイル共有ソフトウェアは効率よくファイルを検索・取得できるとされる．さらに，起動時の挙動に直接的に影響する初期ノードの取り扱いの観点から，通信相手先となるピア数が多いことや，インデックスサーバでファイルの所在を得ることが，効率を上げる要因となっている．手動登録において，初期ノード登録を 1 個のみとしても，比較的短時間で急激にピア数が増加する．これは，自動的に追加されるノードの中に，すでに多くのピアと接続しているノードがあると，初期ノード数に関わらずピア数が増加するためである．しかもユーザはこれを制御できない仕様になっている．

2.2.3　P2P を利用したソフトウェア

ファイル共有ソフトウェア

ファイル共有ソフトウェアは，不特定多数の利用者とファイルを共有するソフトウェアである．表 2.1 は，本論文の実験で対象とした 15 種のファイル共有ソフトウェアである．これらのソフトウェアは，一般社団法人コンテンツ海外流通促進機構が平成 23 年 1 月に公開した調査 [9] をもとに選出されている．概要において型式を明記していないソフトウェアに対して，Gnutella や BitTorrent 系クライアントはハイブリッド型，残りはピュア型を指す．併せて，2.2.2 節で述べた起動時の挙動を指す初期ノードの取り扱いを付記する．

BitTorrent 系クライアントは，torrent ファイルを初期に適用することから，これを初期ノードの手動登録として記載している．

Skype

ファイル共有ソフトウェア以外にも，P2P のスケーラビリティの高さや耐障害性を利用したソフトウェアが多数存在する．例えば，IP 電話，グループウェア，分散コンピューティングなどである．この中でも Skype は，急速に利用が広まっている．Skype は，スーパーノード型 P2P による IP 電話機能を提供している．Skype の会員登録者数は 2010 年 6 月時点で 5.6 億人となっている．

Skype は起動時に P2P ネットワークを構築するため，2.2.2 節で示すような，通信相手

表 2.1　実験対象のファイル共有ソフトウェア

ソフト名	概要	初期ノード
Winny	ピュア型 P2P を利用	手動
Share_EX	Share の TCP 版	手動
Share_NT	Share の UDP 版	手動
PerfectDark	Winny や Share の後継	手動
Cabos	LimeWire の後継	自動
KaZaA	スーパーノード型 P2P を利用	自動
eMule	eDonkey の後継	自動
WinMX	ハイブリッド型 P2P を利用	インデックス
BitComet	BitTorrent クライアント	手動
Shareaza	Gnutella クライアント	自動
Vuze	BitTorrent クライアント	手動
Winnyp	Winny の後継	手動
迅雷	中国発のソフトウェア	自動
μ Torrent	BitTorrent クライアント	手動
StealthNet	端末間通信に AES を使用	自動

先となるホスト数が多い点で，ファイル共有ソフトウェアと挙動が酷似している．本研究では，利用者数の非常に多い Skype を無視できない有用な非ファイル共有ソフトウェアの代表としてとらえ，Skype を誤検出せずにファイル共有ソフトウェアのみを検出する手法を提案する．

2.3　提案手法

本研究では，各種ファイル共有ソフトウェアを個々に識別するのではなく，ファイル共有ソフトウェアとして起動時に検出することを目的としている．そのため，各種ファイル共有ソフトウェアの各挙動を総合的に理解しておく必要がある．

表 2.2　TCP 非ファイル共有

分類	割合
A	0.00
B	88.1
C	5.44
D	6.46

表 2.3　UDP 非ファイル共有

分類	割合
A	0.26
B	45.3
C	0.14
D	54.3

2.3.1　トラヒックパターン

本論文では，ある端末の一定時間内における通信全体を「トラヒックパターン」と呼称する．通信の伝送単位であるパケットは，ある端末からある端末までの通信に必要なヘッダと，通信内容であるペイロードで構成される．

本提案手法では，ペイロードを除いた TCP・UDP/IP ヘッダの情報をトラヒックパターンの分析に利用する．ヘッダ情報のみをトラヒックパターンの分析に用いることで，ネットワーク利用者のプライバシに配慮する．ファイル共有ソフトウェア特有のトラヒックパターンを用いて，ファイル共有ソフトウェアと非ファイル共有ソフトウェアの通信を分別する．

2.3.2　調査実験

ファイル共有ソフトウェアのトラヒックパターンを調べるために，調査実験を行った．調査実験では，前章で述べた 15 種のファイル共有ソフトウェアと Skype を 10 分間動作させ，これらのソフトウェアのパケットを 10 回収集した．以下は，調査実験の結果に基づき，本提案手法で着目したトラヒックパターンである．

■すべてのパケットが TCP か UDP の通信　調査実験で確認した中では，下位レイヤのトンネル・プロトコル（IPSec や L2TP など）を用いる事例は存在しない．これは，通信相手先にも同様の機能が求められるためである．したがって，本論文では，TCP と UDP のトラヒックパターンから提案手法を検討する．

■非 well-known ポートの通信が大半　TCP と UDP の通信を送受信するためのポート番号が well-known ポート番号かどうかで，4 つの組み合わせが存在する．ここで，送信

表 2.4　TCP ファイル共有

分類	割合
A	0.00
B	0.50
C	0.00
D	99.5

表 2.5　UDP ファイル共有

分類	割合
A	0.00
B	0.10
C	0.00
D	99.9

表 2.6　TCP 通信相手先数

ソフト名	宛先ホスト数
PerfectDark	138.2
Share_EX	117.7
Winny	107.8
Shareaza	83.0
Vuze	74.0
Winnyp	53.5
KaZaA	54.2
Skype	8.9
一般トラヒック	0.49

元と宛先のポート番号がともに well-known の通信を A，宛先ポート番号だけ well-known の通信を B，送信元ポート番号だけ well-known の通信を C，送信元と宛先のポート番号がともに非 well-known の通信を D とする．

表 2.2 と表 2.3 は，平常時における和歌山大学内ネットワークからインターネットへの通信全体を分類したものである．表 2.4 と表 2.5 は，ファイル共有ソフトウェアを 10 分間起動させた端末の通信を分類したものである．表 2.2 と表 2.4，表 2.3 と表 2.5 を比較すると，ファイル共有ソフトウェアの通信では，非 well-known ポート番号同士の通信が極めて多く，全体の 99.5 %を占めていることがわかる．

■多くのホストと通信　表 2.6 と表 2.7 は，各ソフトウェアを 10 分間動作させたときにおける，非 well-known ポート通信の平均宛先ホスト数である．ファイル共有ソフトウェアは非 well-known ポートで多くのホストと通信している．

表 2.7　UDP 通信相手先数

ソフト名	宛先ホスト数
BitComet	2705.0
Vuze	772.3
μ Torrent	239.8
eMule	120.0
Share_NT	67.1
Cabos	52.3
KaZaA	40.9
Skype	22.3
一般トラヒック	1.1

表 2.8　初期ノード数とポート番号による相手先数

ソフト名	n-w	n-n	ポート番号	初期ノード数
Winny	4.45	116	80	1
Winny	4.45	114.95	80	400
Share_EX	14.2	174.7	80	1
Winnyp	3.1	55.3	80	1

■初期ノード数とポート番号　初期ノード数と待ち受けポート番号の登録内容によって，宛先ホスト数および非 well-known ポート番号を用いたトラヒックパターンに変化があるかどうか実験した．待ち受けポート番号とは，他のユーザが自端末で稼働するファイル共有ソフトウェアへアクセスする際のポート番号を指す．手動で待ち受けポート番号を設定するファイル共有ソフトウェアは少ない．しかし少ない初期ノード数や，well-known ポート番号で待ち受けた場合でも，通信相手先数の多さと非 well-known ポート番号の高い使用頻度という特性が維持されるならば，これらは利用検知に有効であるといえる．表 2.8 に結果を示す．各ケースは 10 回ずつの起動による宛先ホスト数の平均を示している．表中にある n-w は非 well-known ポート番号から well-known ポート番号への通信相手先数の平均であり，n-n は非 well-known ポート番号同士の同平均を指す．

表 2.8 より，まず初期ノードを最小限の 1 個にしたり，待ち受けを well-known ポート番号にできるファイル共有ソフトウェアであっても，先の第 2 項目における表 2.6 と表

2.7 と同じく，非 well-known ポート番号の利用および宛先ホスト数に変化がないことが確認された．さらに表 2.8 における n-w および n-n の数字により，待ち受けに well-known ポート番号を設定しても，宛先ポート番号に well-known ポート番号を利用する端末からの通信は，非 well-known ポート番号同士の通信の 10 %以下であるといえる．

すなわち，前項と併せて考えれば，ファイル共有ソフトウェア利用検知のため宛先ホスト数に閾値を設定する際，送信元と宛先ポート番号が非 well-known ポート番号同士の通信を対象にすればよいといえる．

■TCP 送信元ポート番号の連続性　アプリケーションが外部と通信する場合，ソケットを作成して送信元ポート番号を選択する．初期の送信元ポート番号は，端末の OS 上でアプリケーション毎に非 well-known ポート番号からランダムに選択される [10]．選択される当該ポート番号はエフェメラルポート（Ephemeral Port, 短命ポート）と呼ばれる．エフェメラルポートは端末の OS 上に実装されており，以下の 2 点を配慮した仕様となっている．1 つ目は複数のアプリケーション間で初期の送信元ポート番号が重複や近接しないことである．2 つ目は同一アプリケーションによる，異なる通信相手先への連続した TCP セッションで，送信元ポート番号が 1 つずつインクリメントされることである [10]．なお，TCP セッション数はキャプチャした TCP の SYN パケット数と等価とする．

すなわち，あるアプリケーションの通信開始時の送信元ポート番号を X とした場合，X～X+10 の区間において，異なる通信相手先への TCP セッション数が 10 に近づくと連続性が高いといえる．逆に，複数の TCP セッションにおける送信元ポート番号の連続性が高ければ，これらは同一のアプリケーションによるトラヒックパターンである可能性が高いといえる．

昨今普及しているタブ型ブラウザなども上記のエフェメラルポートを用いた実装を採用しているため，類似のトラヒックパターンを示すと推測できる．すなわち，上記のポート番号区間を大きくすることと，送信元と宛先ポート番号が非 well-known であることを利用すれば，ファイル共有ソフトウェアとの挙動を区別できる．

筆者らはこれまでに，連続性を判定する上で適切なポート番号区間を決定するための実験を行った [11]．当該実験の目的は，TCP ポート番号の連続性を確認するために，適切な区間の大きさを決定することにより，一般トラヒックとファイル共有ソフトウェアのトラヒックを分離し易くすることである．同一の送信元 IP アドレスに着目した場合，連続したポート番号の個数（区間）の中で，セッション数がどの程度あるかを実験した結果を図 2.1 に示す．なお，実験機 OS には Microsoft® WindowsXP を採用しており，同 OS は送信元ポート番号である非 well-known ポート番号に 1025-5000 を用いている [12]．

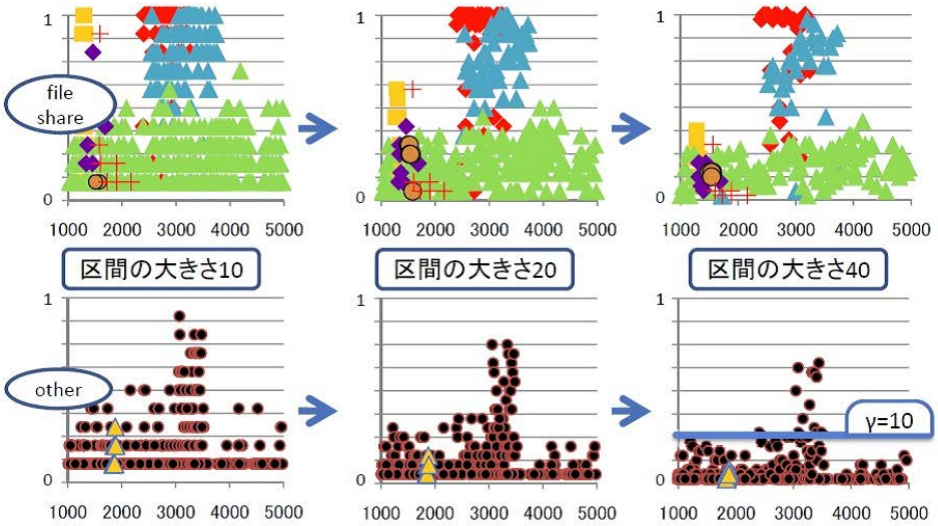


図 2.1　区間の大きさと送信元ポート番号の連続性

図 2.1 は，異なる区間の大きさに対し，送信元ポート番号を連続して使用したヒストグラムを正規化した結果である．上段が各種ファイル共有ソフトウェア，下段が一般トラヒックを指す．横軸は送信元ポート番号である非 well-known ポートを区間の大きさ（10,20,40）で細分化した場合の単位変換された数直線を表している．縦軸は，細分化した区間の中に含まれる TCP セッション数を正規化している．具体的には，同一アプリケーションにおいて，区間 10 に含まれる TCP セッション数が 10 個ある場合，正規値 1 として完全な連続性を示すとともに，グラフの最上位に描点される．

区間が広がると，ファイル共有ソフトウェアより通信相手先数が少ない一般トラヒックは，同区間に含まれるセッション数が減るので，描点がグラフの下方に集まる．一方で，ピュア型のファイル共有ソフトウェアは通信相手先数が多いために，区間が広がっても描点がグラフの上方に集まっている．

図 2.1 と前述の正規値について，送信元ポート番号数である区間 40 における，セッション数が 40 検出された場合，この正規値は 1 となる．したがって，一般トラヒックに着目すると，区間 40 に対してセッション数の閾値 10，正規値 0.25 の場合，送信元ポート番号全域にわたって，その多くを分離できると考えられるが，ファイル共有ソフトウェアのトラヒックは，通信相手先数の少ない同ソフトウェアの描点群に入ってしまう．

したがって，筆者らは，区間を 40 とした場合のセッション数の閾値 20，正規値 0.5 を用いて，連続性を確認することとした．

送信元ポート番号の範囲は OS の実装に依存する．しかし TCP 通信の場合，同一アプリケーションからの連続した TCP セッションで，送信元ポート番号が連続する実装は共通している．これは，エフェメラルポート（短命ポート）を，短時間に複数のアプリケーションで重複利用させないためである [10].

■UDP 送信元ポート番号の同一性　UDP 通信を用いたファイル共有ソフトウェアでは，非 well-known の送信元ポート番号が用いられる．さらに，通信相手先が広く分散しているにも関わらず，同一のファイル共有ソフトウェアは同一の UDP 送信元ポートを使用する．これは，DNS（名前解決）サーバへの問い合わせに比べて，違いが顕著な挙動である．しかし，昨今の Skype による情報共有や事前のシグナリングなどと混同してしまうことが多い.

■一部のソフトウェアが特定ホストへ必ず通信　インデックス型や一部の自動登録型のファイル共有ソフトウェア利用では，起動時に必ず特定のホストに通信することがわかった．対象となるソフトウェアは Cabos，WinMX，迅雷，StealthNet，および Skype である．これは，新旧バージョン間のソフトウェア相互利用をスムーズにするためと推測される.

上記の調査実験で得られた各項目のうち，通信相手先数の多さと非 well-known ポート番号の使用はよく知られている [11][13]．しかし，同一の端末上で複数アプリケーションが併用される状況下では，検出が困難だとされてきた [13]．本研究では，TCP 送信元ポート番号の連続性，および UDP 送信元ポート番号の同一性に着目して，同一端末上でファイル共有ソフトウェアのみの利用検出を試みる．次節では本研究で提案する検出手法について述べる.

2.3.3　提案する検出手法

TCP のトラヒックパターン

調査実験の結果に基づき，ファイル共有ソフトウェアの分類に有効な指標として，本提案手法で着目した TCP のトラヒックパターンについて詳述する．図 2.2 は，ファイル共有ソフトウェアを使用する 1 つの端末が通信するホスト数のグラフである．この際の通信は，送信元と宛先のポート番号として，ともに非 well-known ポート番号を用いた通信に限定している．縦軸は，通信するホスト数であり，ホスト数が 100 以上のものは 100 として表記している．横軸は，ソフトウェアの種別を表しており，種別ごとに 5 回ずつ実験した結果が併記されている．横軸にある一般トラヒックは，和歌山大学内ネットワークで日

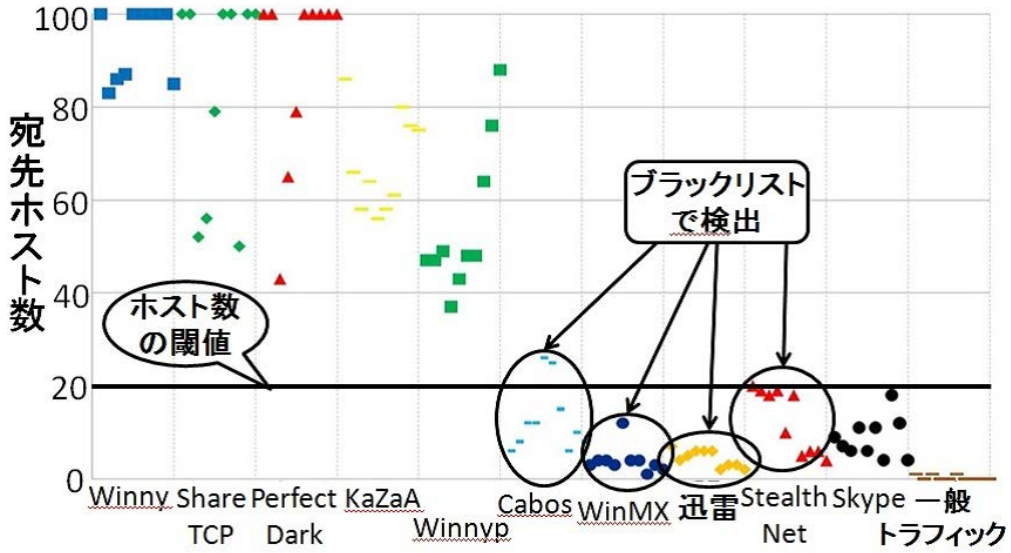


図 2.2　TCP のソフトウェア別通信ホスト数

中に収集したトラヒックである．

Winny，Share，PerfectDark，KaZaA，Winnyp に対しては，通信するホスト数が多いために，通信相手先数に閾値を定めることで，検出可能である．

Cabos，WinMX，迅雷，StealthNet に対しては，Skype や一般トラヒックと同様に，通信の宛先ホスト数が少ないため，通信相手先数では検出が困難である．しかし，通信するホスト数が少ないこれらのソフトウェアには，起動時に特定のホストに必ず通信することが発見された．そこで本提案手法では，これらのソフトウェアに対して，FQDN(Fully Qualified Domain Name) 表記による通信先ホスト名，プロトコルおよびポート番号の 3 項目からなるブラックリストを作成し，ファイル共有ソフトウェアの判別に利用する．いくつかのファイル共有ソフトウェアでは，ユーザインタフェース内に FQDN 表記で特定サーバが示されていた．これは，IP アドレスで指定した場合，サーバの IP アドレスが変わると，全世界の利用者にバージョンアップが必要となるためと推測できる．

表 2.9 は，調査実験で抽出したトラヒックパターンから作成したブラックリストの一部である．通信先のホスト数が少ないファイル共有ソフトウェアは，ブラックリストと TCP 送信元ポート番号の連続性を用いて検出可能である．

さらに，ファイル共有ソフトウェアでは，非 well-known ポート番号を用いた通信が 9 割以上であるために，さらに判定精度を向上できる．

表 2.9　ブラックリストの一部

Software	FQDN	Protocol	Port
WinMX	mainbor.winmxworld.com	TCP	80
迅雷	static.gougou.com	TCP	80
Cabos	gwc.glucolene.com	TCP	8080
StealthNet	webcache.stealthnet.at	TCP	80

表 2.10　ホワイトリストの一部

Software	FQDN	Protocol	Port
Skype	ui.skype.com	TCP	80

TCP での検出手法

前項の図 2.2 の通り，通信の宛先ホスト数のみによって，Cabos，WinMX，迅雷，StealthNet のような通信相手先数の少ないファイル共有ソフトウェアと，Skype や一般トラヒックとを区別することは困難である．そこで本提案手法では，TCP トラヒックを対象として，通信の宛先ホスト数の多いファイル共有ソフトウェアに対しては，宛先ホスト数の閾値を用いて検出を行う．図 2.2 より，宛先ホスト数の多いファイル共有ソフトウェアと一般トラヒックを分類するための当該閾値を，図中に示したように 20 とする．一方で，通信の宛先ホスト数の少ないファイル共有ソフトウェアに対して，1.「ブラックリスト」，2.「非 well-known な宛先ポート番号の通信数」および 3.「TCP 送信元ポート番号の連続性」の手順を用いて検出を行う．

UDP のトラヒックパターン

調査実験の結果から，分類指標として着目した UDP のトラヒックパターンについて詳述する．図 2.3 は，ファイル共有ソフトウェアを使用する 1 つの端末が通信するホスト数のグラフである．この際の通信は，送信元と宛先のポート番号が well-known ポートでない通信に限定している．グラフの記述方法は図 2.2 と同様である．

図 2.3 より，ファイル共有ソフトウェアと一般トラヒックを判別することは可能である．しかし，通信の宛先ホスト数だけでは Skype を誤検出してしまう．調査実験の結果，Skype では起動時に特定のホストに必ず通信することが判明した．そこで本提案手法で

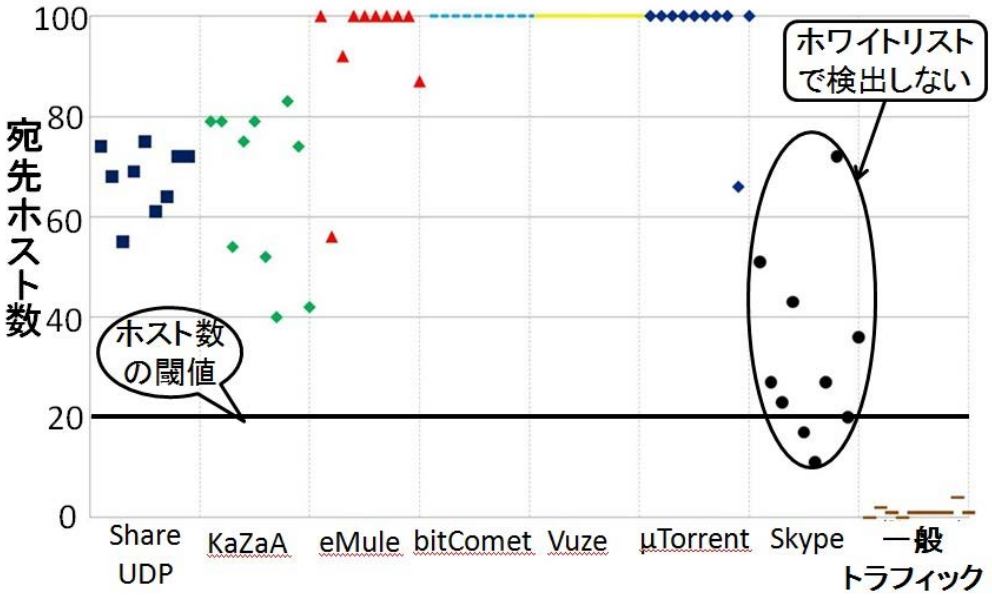


図 2.3　UDP のソフトウェア別通信ホスト数

は，Skype に対して FQDN 表記による通信先ホスト名，プロトコルおよびポート番号の 3 項目からなるホワイトリストを作成し，Skype の判別に利用する．表 2.10 は，作成したホワイトリストの一部である．

UDP によるファイル共有ソフトウェアの場合，同一のソフトウェアによる通信では，送信元ポート番号が常に同じ値になることが事前実験より判明した．したがって，同一の送信元ポート番号からなる UDP 通信は，すべて同一のファイル共有ソフトウェアからの通信であると推測できる．

一方で，Skype の場合は同一の送信元ポート番号からなる UDP 通信相手先数が，ファイル共有ソフトウェアに比べると少ないことが実験より判明している．

UDP での検出手法

前項の図 2.3 の通り，通信の宛先ホスト数のみによってファイル共有ソフトウェアと Skype との区別は困難である．そこで本提案手法では，UDP トラヒックを対象として，1.「ホワイトリスト」，2.「UDP 送信元ポート番号の同一性」および 3.「非 well-known ポート通信の数」の手順を用いて，UDP を用いたファイル共有ソフトウェア利用を検出し，Skype の誤検出を防ぐ．同一の送信元 IP アドレスに対して，同一の送信元ポート番号を有する通信および非 well-known ポート番号を用いた通信を抽出する．抽出された通

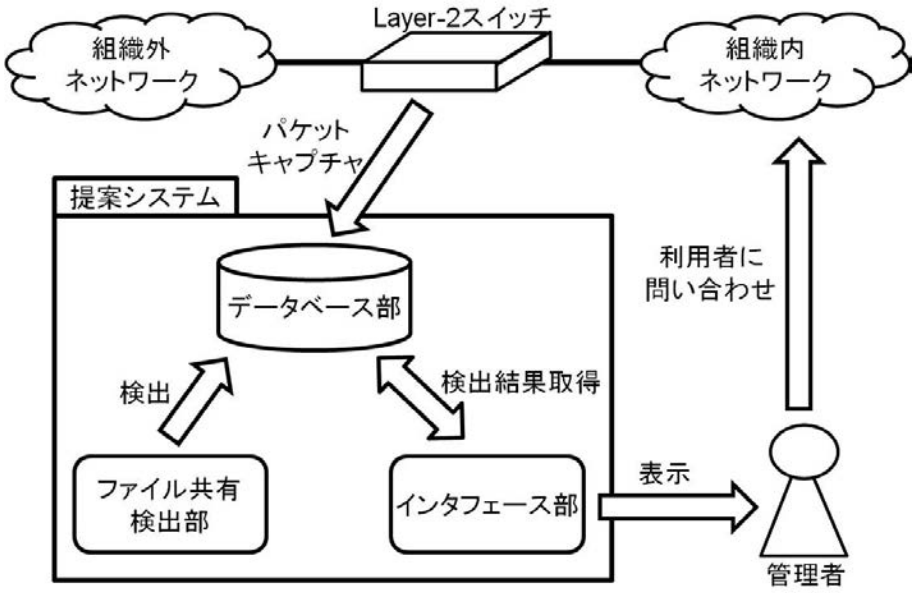


図 2.4　提案システムの構成図

信の中で，異なる宛先 IP アドレスの数の総和が，UDP によるファイル共有ソフトウェアの検出に必要な通信相手先数となる．この閾値を 20 とする．

前節のように，Skype は，上記に示した UDP による通信相手先数が少ないので除外できる．しかし，同一端末内で Skype とファイル共有ソフトウェアを稼働させた場合を想定して，ホワイトリストを採用する．

2.3.4　提案システム

提案システムは，提案手法を FreeBSD 上に実装した．ハードウェア仕様としては，Intel 社製 Core i5，メモリ 2GB および 1TB の HDD である．データを格納するデータベースとして MySQL，Web ブラウザから本システムを管理するための Web Server として Apache を使用した．

本システムの構成図を図 2.4 に示す．本システムは組織内ネットワークと組織外ネットワークの境界部分への導入を想定している．本システムは，データベース部，ファイル共有検出部，インタフェース部の 3 つに大別できる．データベース部は，プロミスキャス (promiscuous) モードで動作するネットワークインタフェースを通じてパケットをキャプチャする．そして，キャプチャしたパケットの TCP・UDP/IP ヘッダの情報をデータベースに格納する．ファイル共有検出部は，提案手法を用いてファイル共有ソフトウェア

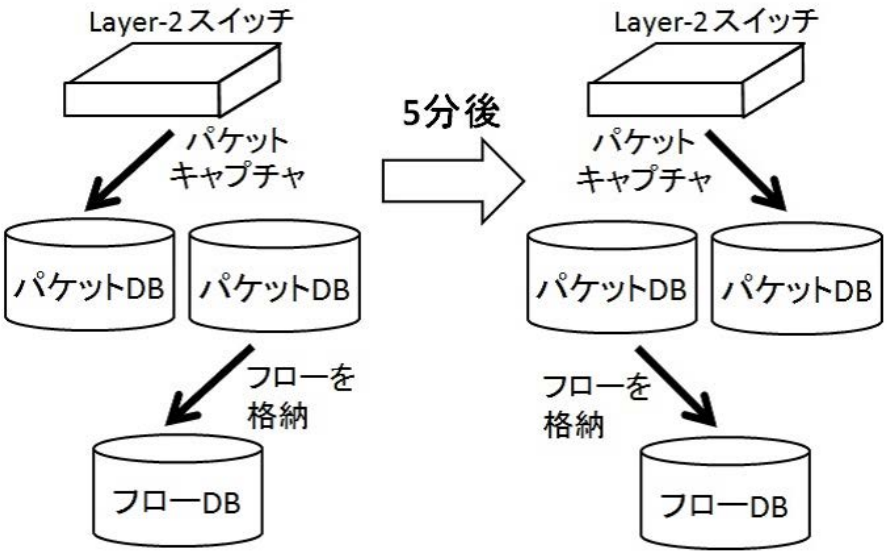


図 2.5　パケット DB とフロー DB

を検出する．インタフェース部は，管理者に提案システムの動作状況とファイル共有ソフトウェアの検出結果を表示する．

データベース部について詳述する．データベース部には 2 種類のデータベースを実装している．キャプチャしたパケットを格納するパケットデータベース（以下パケット DB），および TCP あるいは UDP による同一通信のパケット群をまとめたフローデータベース（以下フロー DB）である．パケット DB はパケット数を高速に処理するためメモリ上にあり，フロー DB は膨大な記録を保存するため HDD 上に実装されている．図 2.5 に示すように，パケット DB は 2 つあり，5 分ごとに交互にパケットをキャプチャしパケット DB に格納する．パケット DB がパケットをキャプチャする間隔には，5 分間の観測後に利用を検出する既存研究 [14] を参考にした．実際には，ファイル共有ソフトウェアが起動直後に P2P ネットワークを構築するのに数十秒かかる．しかし，これは初期ノードの質，すなわち指定する初期ノードが多くの既接続ピアをもっているかに依存する．さらに，本研究は利用検知が目的であり，ファイル共有ソフトウェア起動直後の停止措置を必要としない．以上の考察と前述の事前実験の結果から，本論文ではパケット DB が切り替わる時間間隔に 5 分間を採用する．

そして，パケットをキャプチャしない時間帯において交互にフローデータを作成してフロー DB に格納する．フローデータとは，同一の送信元および宛先の IP アドレスとポート番号によって識別されるパケット系列を同一の通信（フロー）と判断し，これを整理し

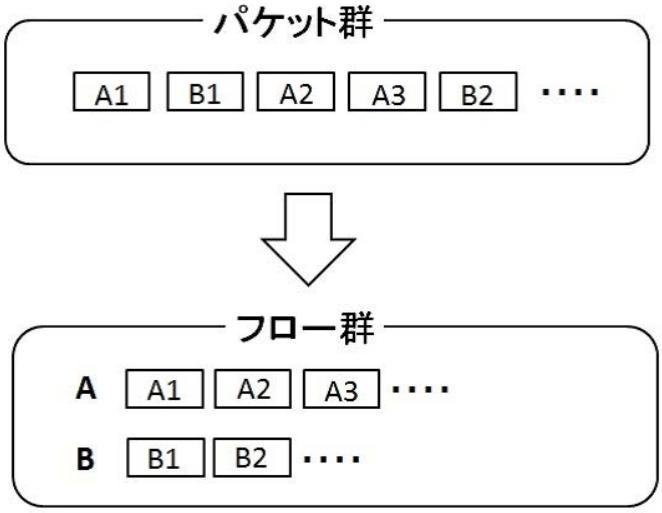


図 2.6　フローデータのイメージ

たデータを指す．図 2.6 に，フロー DB に格納する際のフローデータのイメージを示す．1 つの送信元 IP アドレスに着目した場合，宛先 IP アドレスと宛先ポート番号が一致するパケット A およびパケット B について，パケットデータは図 2.6 上段のようになり，これらを取得順にパケット DB に格納する．しかし，キャプチャと並行してパケット DB 内で通信相手先数やポート番号の連続性・同一性を検出するには，パケット DB のデータ全域を走査する必要があり効率が悪い．

2.3.1 節で述べたように，必要な要素は同一の送信元 IP アドレスからのトラヒックパターンである．したがって，これを基点にした通信相手先数と送信元ポート番号の連続性・同一性を確認すればよい．フロー DB に格納する際，図 2.6 下段に示すように，まず送信元アドレスと宛先アドレス，および送信元，宛先ポート番号の違いに基づき個々のフローエントリを作成する．5 分間でキャプチャしたすべてのフローエントリを作成した後に，各フローエントリを送信元ポート番号の小さい順に並べ替えて全体のフローデータとする．これは，後述する図 2.7 の右上にあるように，一つの送信元アドレスに着目した際のフロー群を抽出しやすくするためである．

本研究におけるファイル共有ソフトウェアの検出に必要な情報は，IP アドレス（通信相手先数）とポート番号（連続性・同一性）などであり，これらはフローデータに格納されている．したがって，検出時の検索負荷を下げるため，ファイル共有検出部とインタフェース部はパケット DB ではなく，フロー DB を用いることとした．

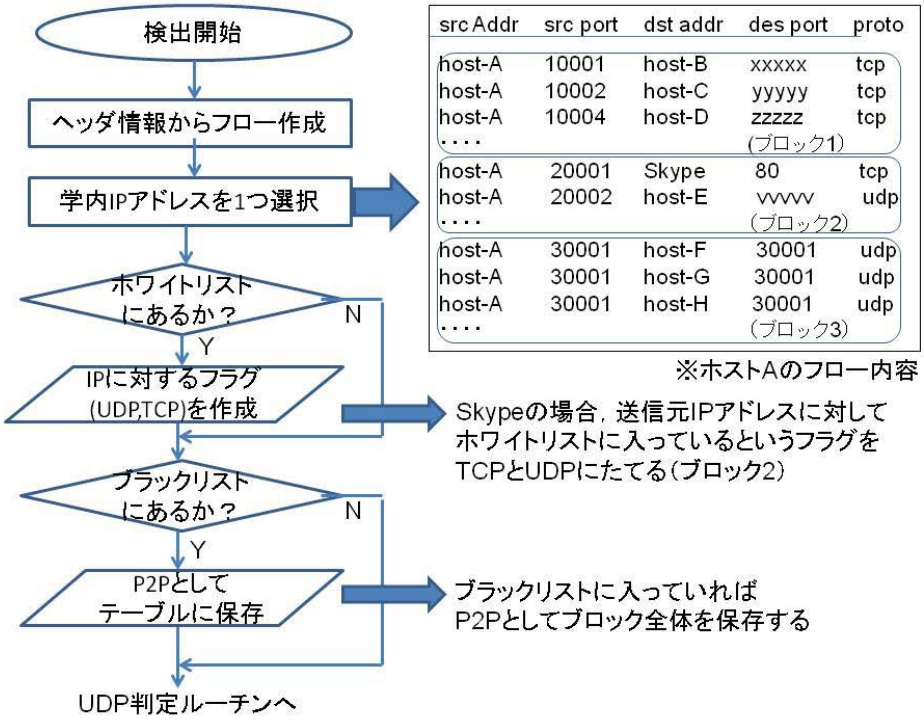


図 2.7　リストによる判定ルーチン

なお，ネットワーク利用の統計情報を取得する手法として，NetFlow に対応したネットワーク機器の使用がある [15]．NetFlow により取得できる統計情報には，送信元および宛先の IP アドレス，ポート番号など多くの項目が存在し，それらは本提案手法におけるフロー DB に格納する情報として利用できる．しかし，すべてのフローデータを取得するには，高性能のネットワーク機器が必要であり，サンプル抽出したフローデータでは，本提案手法にとって有用とならない．詳細を 2.4.2 節で述べる．

2.3.5　利用検出フローチャート

送信元 IP アドレスを基点にして，5 分間のフロー DB から同一ホストのフローを抽出して利用検知を実施する．同一ホストで複数のアプリケーションが稼働している場合，図 2.7 右上のように，フローの中に複数のブロックが存在する．ブロックとは，2.3.2 節で述べたポート番号の連続性や同一性の検知に用いやすくするため，送信元ポート番号順にフローを整理した上でのフローの集合体である．すなわち，近接した送信元ポート番号は，同じアプリケーションからの通信であることから，これを基準にして各ブロックに分

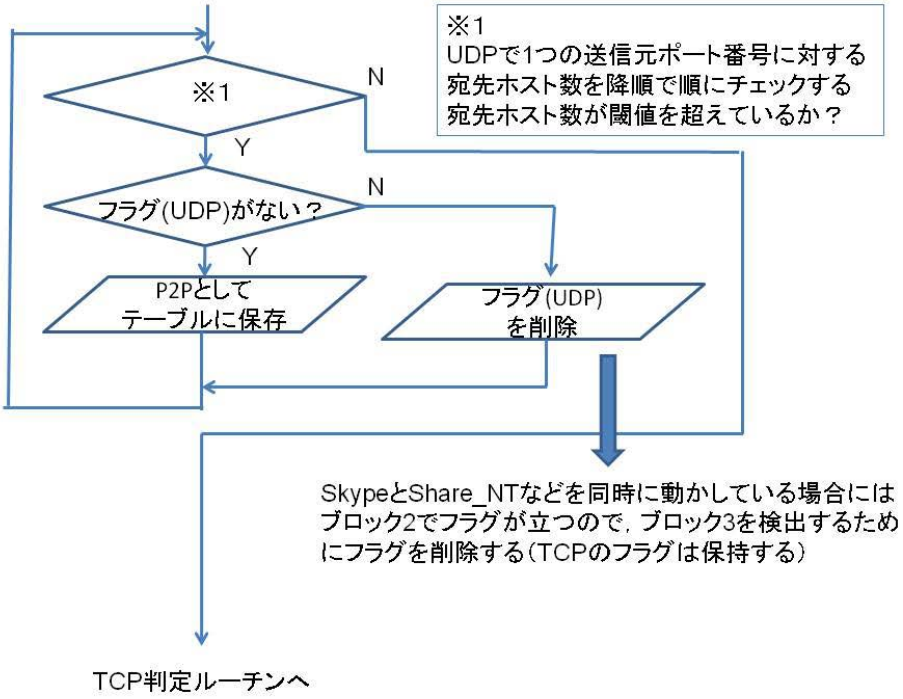


図 2.8　UDP 判定ルーチン

けることで判定に用いる．なお，5 分間の観測時間内に複数のアプリケーションが同一の送信元ポート番号を使用する事例は，実験では確認されていない．したがって，送信元ポート番号が 100 以上離れている場合，ブロックを切り分けることとする．これは．エフェメラルポートの仕様に沿った送信元ポート番号が，アプリケーションごとに離れた間隔でランダム選択されているためである．以降の処理におけるブラックリスト，UDP および TCP 判定ルーチンは，同一の送信元 IP アドレスにおけるブロックごとに実施される．

ブラックリスト・ホワイトリスト

図 2.7 におけるブロック 1 およびブロック 3 は，TCP と UDP によるファイル共有ソフトウェアのトラヒックパターンであり，ブロック 2 は Skype のトラヒックパターンである．リストによる判定では，リストに含まれる特定サーバの FQDN を DNS で正引きし，得られた IP アドレスあるいは IP アドレス群と宛先 IP アドレスを比較する．

宛先 IP アドレスがホワイトリストに含まれる場合，当該送信元 IP アドレスに対応するフローエントリに，後に TCP 判定ルーチンで検査するためのフラグ（以下，TCP フラ

グ）と TCP 判定ルーチンで検査するためのフラグ（以下，UDP フラグ）を立てる．これは図 2.7 のフロー例のように，ホワイトリストに含まれるホストであっても，同時に別のファイル共有ソフトウェアを実行している可能性があるためである．図 2.7 のフロー例では，ブロック 1 は後述する TCP 判定ルーチンで検出される．ブロック 2 は Skype をホワイトリストによって除外できるが，ブロック 3 に同一の送信元 IP アドレスが含まれており，Skype と誤検知されないよう配慮する必要がある．このため，ブロック 2 で TCP フラグと UDP フラグを立てた後，UDP 判定ルーチンと TCP 判定ルーチンのどちらかにファイル共有ソフトウェアのトラヒックパターンを確認できた場合，該当するフラグを削除して再検査する．これにより，ブロック 3 を Skype と誤検知することなく検知する．

宛先 IP アドレスがブラックリストに含まれる場合は，送信元 IP アドレスのフロー全体の中から，当該通信を含むブロックを，ファイル共有ソフトウェアを利用検知した結果として検知テーブルに保存する．検知テーブルとは，フロー DB の中から該当するフロー群であるブロックを記録する別のデータベースであり，同テーブルに保存後，次のブロックに対して以降の処理を継続する．

上記の当該ブロックは，近接する送信元ポート番号を用いた通信のうち，送信元と宛先が非 well-known ポート番号のものが含まれている場合に利用検知する．これは，ブラックリストに含まれる通信のみで検出する場合，表 2.9 の宛先ポート番号が 80 番などであることから，単なる Web 閲覧を false-positive で検出してしまう可能性があるためである．例として，WinMX では，ブラックリストに含まれる通信の検査の後に，非 well-known ポート番号を用いた 3-5 個のセッションが確認された．なお，送信元ポート番号が近接する非 well-known ポート番号同士のフローエントリは，ブラックリストに合致する同エントリの前後 50 から選択する．本来ブラックリストに記載するファイル共有ソフトウェアは通信相手先数が少ないため，実験では上記選択範囲の中で検知できている．

図 2.7 において，ホワイトリストとブラックリストによる判定ルーチン以降の UDP および TCP 判定ルーチンでは，送信元および宛先が非 well-known ポート番号を有するフローエントリを対象とする．これは，2.3.2 節および 2.3.3 節に基づく設計である．

UDP 判定ルーチン

次に，UDP 判定ルーチンでは，図 2.8 に示すように，ポート番号の同一性と通信相手先数により利用を検知する．図 2.8 の初期段階で，同一の送信元ポート番号を用いた通信相手先数をカウントし，ブロック 3 を検出する．その後，図 2.7 で判定したホワイトリストでの UDP フラグがないか判別する．フラグがあれば，これを削除して再度ブロック 3

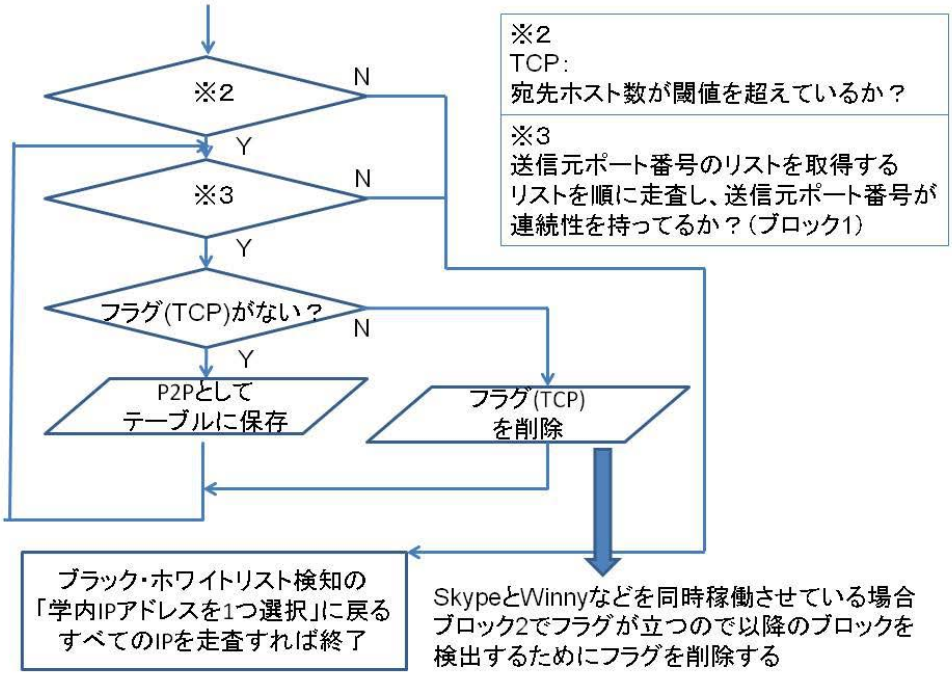


図 2.9　TCP 判定ルーチン

を検出し，同ブロックを検知テーブルに保存し，以降の処理を継続する．

フラグを削除して再度通信相手先数を判定することは，Skype をファイル共有ソフトウェアとして検出する false-positive を確認するための仕様である．本フローチャートで Skype の false-positive が発生する箇所はフラグを削除する部分であり，フローエントリと検知テーブル，および端末側で稼働している Skype の通信ログとの比較で false-positive を確認する．

2.3.3 節で述べたように，Skype は同一の送信元ポート番号と非 well-known ポート通信による相手先数が少ないため，UDP 判定ルーチンではブロック 2 を検知しない．Skype で同時に閾値 20 以上の相手先と通話を開始すると検知される可能性はあるが，現在では一般的な利用形態ではないため，本研究では除外する．

TCP 判定ルーチン

最後に，TCP 判定ルーチンでは，図 2.9 に示すように，通信相手先数とポート番号の連続性により利用を検知する．2.3.2 節で述べたように，ポート連続性は区間 40 に含まれる TCP セッション数 20 を閾値として検出する．同じく 2.3.2 節で述べたように，

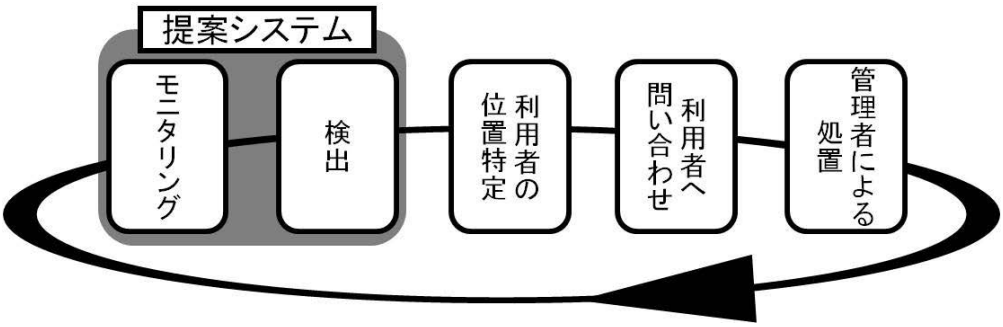


図 2.10　運用フレームワーク

well-known ポート番号を宛先として設定しても，そのセッション数は非 well-known ポート番号同士の通信の 10 %以下であるため，連続性を検出できる．

2.3.6　運用フレームワーク

前節で，本論文で提案するファイル共有ソフトウェアの検出手法について述べた．しかし，提案手法によって，ファイル共有ソフトウェアに似た挙動の非ファイル共有ソフトウェアを誤検知する可能性は存在する．

そこで，本研究では提案手法を実装したシステムを用いた運用フレームワーク（図 2.10 参照）を提案する．提案システムは，通信のモニタリングとファイル共有ソフトウェアの検出を担当する．ファイル共有ソフトウェアの利用者を特定すると，管理者から利用者に問い合わせる．利用者がファイル共有ソフトウェアを利用していないと確認できた場合，当該利用者の端末のトラヒックパターンを解析し，提案システムにフィードバックする．このような循環型の運用で，提案システムの検出精度をより向上させることが可能である．

具体的には，一般トラヒックや Skype を false-positive によって誤検知した場合，TCP であれば通信相手先数の閾値やポート連続性の閾値を上げて再稼働させるなどがある．一方で，false-negative は通常検出できないため，ファイル共有ソフトウェアのバージョンアップなどがあった場合，上記の閾値で検知可能かどうか，確認する必要がある．しかし，DPI 型のようにペイロードを解析するほどの負担は不要である．

2.4 評価・考察

2.4.1 評価実験

提案システムを用いた評価実験について述べる．実験の主旨は検出精度を通常トラヒックの中で確認することである．そこで，和歌山大学の実トラヒックが集約される対外接続部分に提案システムを導入した．同部分において，学内側に実験対象のソフトウェアを複数内包する端末を設置して稼働させた．実験対象のソフトウェアは，2.2.3 節で述べた 15 種のファイル共有ソフトウェアと Skype である．端末の処理能力と一般的なファイル共有ソフトウェアの利用形態を考慮して，一台の端末に 1-2 個のファイル共有ソフトウェアおよび Skype を起動させる．前提条件として，ファイル共有ソフトウェア群はダウンロードに必要なファイルの検索およびダウンロードを実施しない．一般的には，ファイルの検索およびダウンロードによって，より顕著に挙動を検出できると推測されるが，将来的に提案システムが当該通信を遮断することを想定して，本研究では起動時の挙動のみで検出を試みる．

和歌山大学の対外接続部では，平日の 5 分間平均で最大 110-120Mbps のトラヒックが発生する．特に 14 時から 16 時は同程度のトラヒックが維持されているため，同時間帯において実験対象のソフトウェアを 10 分間ずつ 5 回起動させた．提案システムを動作させる上で必要になる検出の閾値は，TCP および UDP における通信相手先数の場合，事前調査の結果（図 2.2，図 2.3）から 20 とし，通信相手先数の多いファイル共有ソフトウェアを検出することとした．併せて，送信元ポート番号の連続性の場合，事前の調査実験から区間を 40，セッション数の閾値を 20 として，同一のファイル共有ソフトウェアや同一の Skype による通信を検出することとした..

検出結果として，Skype を同時に起動させた上で，15 種のファイル共有ソフトウェアをすべて検出し，Skype は 1 度も誤検出しなかった．5 分間のパケットキャプチャにおいて，利用検知までの平均時間を表 2.11 に示す．5 分間のパケットキャプチャにおけるパケット DB サイズは，学内のスループットピーク時（双方向で 5 分間の最大平均がおよそ 110Mbps）でおよそ 47MB，フロー DB のサイズはおよそ 4.6MB である．表 2.11 に示すように，組織内でファイル共有ソフトウェアを利用した場合，パケットキャプチャと利用検知には，最長で 5 分とおよそ 11 秒を要することになる．

さらに，2.3.5 節で述べた利用検出フローチャートにおいて，どのルーチンでどのファイル共有ソフトウェアが検知されたかを表 2.12 に示す．

表 2.11　検出までの平均時間

キャプチャ	フロー DB	リスト処理	UDP 判定	TCP 判定
5min	4.5sec	$<$ 1sec	1.5sec	5sec

表 2.12　検知したファイル共有ソフトウェア

ブラックリスト	Cabos, 迅雷,WinMX,StealthNet
UDP 判定ルーチン	Share_NT,KaZaA,eMule,BitComent, Shareaza,Vuze, μ Torrent
TCP 判定ルーチン	Winny,Share_EX,PerfectDark, KaZaA,Shareaza,Winnyp

表 2.12 に示すように，TCP および UDP 双方を使用する KaZaA や Shareaza を各判定ルーチンにて各々検知している．これは，同一の送信元 IP アドレスに対して TCP および UDP の判定ルーチンを適用できることに起因する．さらに，この実験では意図的に Skype を稼働させており，ホワイトリストによる対応によって，同一端末内での複数アプリケーション稼働が，ファイル共有ソフトウェアの利用検知に影響しないことが確認できた．

なお，本提案手法を用いた検出システムでは，ファイル共有ソフトウェアの通信の検出は可能だが，各種のファイル共有ソフトウェアを個別に識別しない．すなわち，本実験における検出とは，ファイル共有ソフトウェアとしての起動時の挙動を検出することを指す．

2.4.2　考察

ブラックリストについて

本論文では，通信の宛先ホスト数とブラックリストを用いた検出を提案した．ブラックリストを用いることで，通信相手先数の少ない同ソフトの検出精度向上と誤検出の削減を実現できた．

ブラックリストを作成する運用上の手間が存在する．しかし，ブラックリスト作成対象である，通信の宛先ホスト数の少ないファイル共有ソフトウェアの利用は減少傾向にある．平成 22 年度のファイル共有ソフトの利用に関する調査 [9] によると，宛先ホスト数

の少ない形式の P2P から宛先ホスト数の多い形式の P2P を利用したソフトウェアに移行するとされている．

本研究では重視しなかったが，一般にハイブリッド型 P2P などは，インデックスサーバが開発サイトに固定されていることが多い．さらに，同型 P2P はインデックスサーバ上で利用者を特定しやすい点で匿名性が低いといえる．ファイル共有サービスが開発された当初，最も利用されていたファイル共有ソフトウェアである Napster や LimeWire は，著作権違反によりインデックスサーバを停止させられることでサービス停止となった [16][17].

昨今の著作権違反に対するファイル共有ソフトウェア利用者の検挙事例により，ユーザはより匿名性の高いファイル共有ソフトウェアを求めて，ハイブリッド型 P2P から離れる傾向にあると考えられる．ユーザが離れたファイル共有ネットワークはコンテンツが不足するため，さらに利用が減少する．このようにして，ハイブリッド型 P2P のファイル共有ソフトウェアの利用は減少傾向にある．

提案システムは，ファイル共有ソフトウェア利用全体の 2 割 [9] に相当する Cabos や WinMX に対してブラックリストを作成した．ハイブリッド型 P2P を利用するソフトウェアの利用が減少することで，ブラックリストを新たに作成する機会は今後減少すると考えられる．なお，一般に BitTorrent 系クライアントはハイブリッド型 P2P とされている．しかし torrent ファイルに含まれるコンテンツ毎にクライアントが群体を成して相互接続するため，インデックスサーバの特定が困難で匿名性が高い．したがって，上記の減少傾向には当てはまらないと考える．

ホワイトリストについて

提案システムでは，同一端末内でファイル共有ソフトウェアと類似の挙動を示す Skype を誤検知しない手法を実装し，実際に検知しないという結果を得た．

前節のブラックリストに比べて，ホワイトリストの必要性は今後増大していくと考えられる．IP 電話，グループウェア，分散コンピューティング，P2P 放送，P2P 通貨など，P2P 通信を利用したサービスは日々生み出されている．上記サービスとファイル共有サービスを区別するためにも，ホワイトリストを利用したシステム運用は必要であると考えられる．さらに，BitTorrent を使用した OS のダウンロードなど，ネットワーク利用者からの様々な要望を想定すると，柔軟なネットワーク運用のために拡張ホワイトリストによる運用は必須であると考える．現在のホワイトリストは，通信相手先とポート番号を指定している．これが組織内の特定 IP アドレスを指定したホワイトリストの場合，以降の判定ルーチンをすべてパスさせてしまう．したがって，上記の特定 IP アドレスを持つ

フローデータを再検証する手法を検討している．

関連研究と提案方式の特徴

ファイル共有ソフトウェアの検出手法は様々な研究で提案されており，製品も存在する．これらは検出手法から，クローリング型，ゲートウェイ型，トラヒックモニタ型の 3 種に分類できる．

クローリング型は，ファイル共有調査端末がファイル共有ネットワークに参加することで情報収集する．しかし，各ファイル共有ネットワークで膨大な情報収集が必要であるために，検出可能なソフトウェアが少なく，一つの企業団体だけでこの手法を導入することは困難である．クローリング型で検出するものに P2P FINDER [18] や P2P 観測システム [19] がある．

ゲートウェイ型は，多くが DPI 方式であり，組織内外のネットワーク境界にゲートウェイ端末を設置し，通過するパケットを逐一チェックして検出・遮断する．ゲートウェイ端末には高いスループットが求められるために，多くは専用機（アプライアンス）がパケットのペイロードを検閲し分類する．ゲートウェイ型の製品として One Point Wall [6] や Palo Alto 社製の統合ファイアウォールである Palo Alto [7] などが存在する．これらは検出用のシグネチャ更新など保守契約を含め非常に高価であり，様々な組織への導入が困難である．この理由には，StealthNet など AES 暗号方式で暗号化された通信では，検出が極めて困難であるため，個別の開発コストが高くなることなどが挙げられる．さらに Winnyp など利用者の分布が主に開発国に限定される場合には，コスト対効果が低く，検知対象とならないことがある．

なお，ペイロードの検閲や価格面で本研究の主旨とははずれるが，Palo Alto を別途導入した実験も行った．その結果，実験時に StealthNet を検出する一方で，Winnyp を検出できなかったことを付記しておく．

トラヒックモニタ型は，組織内ネットワークと組織外ネットワークの境界を流れるパケットをモニタリングすることで検出する．トラヒックモニタ型では，パケットのペイロードを検閲することなくファイル共有ソフトウェアを検出する手法が多数提案されている [14][20][21][22][23] [24].

しかし，トラヒックモニタ型で提案されている検出手法は，Winny など非常に特徴的なピュア型に限定された検出 [20][21][22][23][24] であり，ファイル共有ソフトウェアに対する汎用性が低い．さらに，Skype の誤検知を確認した事例 [14][24] では，解決策が提示されていない．これは，通信相手先数や通信の間隔に着目しているため，通信相手先数が極端に多くない場合の精度に限度があるためと考えられる．本論文の実験では，様々な

ファイル共有ソフトウェア 15 種を検出しているため，汎用性と精度が高いといえる．

一方，検出に必要なサンプリングの時間に着目すると，既存手法では，30 分 [23]，1 日 [24] といったオーダーで測定されるなど，即時性が低いといえる．

Holt-Winters 法を用いた約 1 カ月間の学習 [14] では，学習後 5 分間の観測と 1.2 秒の計算で検出が可能だが，Skype など同一端末での複数アプリケーションの判別を課題としている．本論文の提案手法では，学習期間を必要とせず，同じ 5 分間のパケットキャプチャで Skype との判別が可能である．

上記の各既存手法では，フロー抽出や実環境で実際に検出を試みた事例は極端に少ない．実際の ISP（Internet Service Provider）において，ファイル共有ソフトウェア利用検知を試みた研究 [13] がある．これは，バックボーン OC-48（2.4Gbps）において，概念は少し違うが，筆者らと同じくフローを用いた検知の効率化を図っている．しかし，フローによる走査だけでは，同一端末上での複数アプリケーション稼働からファイル共有ソフトウェアを検知することが困難であり，これを課題として挙げている．

本論文の提案手法では，ブラックリスト・ホワイトリスト，通信相手先数，UDP 送信元ポート番号の同一性と TCP 送信元ポート番号の連続性を用いることで，端末が使用するファイル共有ソフトウェアの検出が可能になった．同時に，本手法ではファイル共有ソフトウェアの起動時の挙動のみで検出可能であり，同一の端末から通常のトラヒックが送受信されている状況下でも，ファイル共有ソフトウェアの利用のみを抽出できる．さらに，和歌山大学内ネットワークにおける評価実験によって，実環境においても提案手法が有効であることが実証された．

フローデータの有用性

評価実験では，データベース部のパケットキャプチャにおいて，取得したパケットのロス値を 2 種類の方法で計測した．1 つ目の方法は，通信経路上のレイヤ 2 スイッチ内のアップリンクポートにおける通過パケット数と，提案システムのネットワークインタフェースでキャプチャしているパケット数の比較により計測する．2 つ目の方法は，和歌山大学の対外接続部における学内と学外に定期的および継続的にデータ転送する端末を別途用意し，データ転送に要したパケット数とパケット DB 内の当該パケット数との比較により計測する．この結果，双方の計測方法において最大で 10 %程度のキャプチャ・ロスが確認された．本評価結果により，当該ロスは検出結果に深刻な影響を与えなかったといえる．他にもフローデータに基づく処理の効果には，キャプチャと検索が別プロセスであること，それぞれの格納場所をメモリ上と HDD 上に分けることで，データベースへの入出力の負荷が抑制されることが挙げられる．

なお，本論文ではパケット DB の切替にともなうロスのみを限定して計測してはいない．パケット DB が稼働するメモリへの書き込み速度は，昨今のメモリ（DDR3）性能により 1.6Gbps 以上である．和歌山大学の対外線の瞬間最大スループットは 600Mbps 程度であり，パケット DB 切替時のロスによる影響は少ないと判断した．しかし，トラヒックの増大によって，今後問題が生じる可能性がある．

さらに，パケット DB の切替時を跨いでトラヒックパターンが分散する場合を考慮して，フロー DB への格納時以外でも，フロー DB に対して定期的に検索をかけるスクリプトを稼働して対応している．

NetFlow によるフローデータ

2.3.4 節で述べたように，NetFlow によるフローデータには，送信元，宛先の IP アドレスおよびポート番号が含まれる．したがって，本提案手法のようなパケット DB を介することなく，直接フロー DB にフローデータを格納する形で利用できると考える．しかし一方で，NetFlow 機能を有効に活用するには，同機能を有するネットワーク機器に相当の性能が求められる．

NetFlow におけるフローは，TCP あるいは UDP 通信の開始時と終了時の時刻を取得するため，終了するまでフローをネットワーク機器上のフローテーブルに保持する．フローテーブルのサイズには限度があり，この限度はネットワーク機器のグレードや内蔵する OS に依存する．2.4.1 節および前節で示した，和歌山大学のような中規模の組織では，組織内ネットワークのコアとなるレイヤ 3 スイッチ相当以上の機器が必要となる．一方で NetFlow には，すべてのフローではなく，時間やパケット数に基づく一部のサンプルフローを取得する機能がある．例として時間に基づく最大サンプリングレートでは，Cisco Systems 社製の場合，4096msec のうち最初の 64msec を取得する．

本提案手法では，NetFlow によるフローデータの取得を実施せず，安価なレイヤ 2 スイッチのモニタポートでパケットをキャプチャした．しかし，前節のようにトラヒックの増大を考慮して，適切なサンプリングレートで検知できるかなど，検討していく必要がある．

今後の課題

文部科学省や IPA のガイドラインでは，ファイル共有ソフトウェアを「できるだけ使わせない」となっているが，悪意の是非を規定することは困難である．したがって，運用フレームワークに基づくヒアリングと，拡張ホワイトリストの開発など現実的な解決策の実装を進める必要がある．併せて，複数アプリケーション同士が 5 分間で送信元ポート番

号を重複させる事象が存在するかどうかの検証や，ホワイトリストのフラグを削除して再度通信相手先数をチェックするという非効率性を低減させる方法など，さらに効率と精度を高めるべく仕様を検証する必要がある．加えて，NetFlow によるフローデータを利用した検知手法とパケット DB の切替時に発生するロス値についても検討する必要がある．

2.5 結言

本研究では，ファイル共有ソフトウェア利用端末のトラヒックパターンを解析することで，同ソフトウェアを検出する手法を提案した．従来の通信相手先数と非 well-known ポートに着目する手法に加えて，送信元ポート番号の連続性や同一性に着目し，ホワイトリスト・ブラックリスト判定と，UDP および TCP 判定により高精度な利用検知を実現した．これらはパケットのペイロードを検閲することなく，トラヒックパターンのみを用いている．さらに，有用な P2P ソフトウェアである Skype については，ホワイトリストで誤検出を防ぐ手法を提案し有効性を確認した．同一端末で複数アプリケーションが稼働する状況下の評価実験を実際の運用ネットワーク上で実施し，一般社団法人コンテンツ海外流通促進機構が平成 23 年 1 月に公開した調査に基づく全 15 種類すべてのファイル共有ソフトウェアの検出を確認するとともに，本研究の有用性を示した．

今後，様々なソフトウェアが開発されることを考慮すると，各リストを用いた柔軟なファイル共有ソフトウェア検出の運用手法は必須である．しかし，新種やバージョンアップされたソフトウェアが現れる都度，管理者の手によってトラヒックパターンを収集して解析するには高い負担が求められる．このため，今後の課題として，継続してトラヒックパターンを収集し効率的に解析する仕組みを構築する必要がある．

加えて本研究では，トラヒックパターンの検知手法を明確に定義して実施している．したがって，今後はパターン認識の観点から，SVM(Support Vector Machine) などの機械学習によって，各ファイル共有ソフトウェアを判別できるシステムへと改良する所存である．

第 3 章

DoS 攻撃に対する透過型防御システムの構築とステートレスな TCP 代理応答の評価

3.1 序言

組織内ネットワークからインターネットに対して提供するサービスが充実するとともに，当該サービスが停止した場合の損失や社会的影響が深刻化している．企業や行政機関，および学術機関を対象とした標的型攻撃の 1 つにサービス妨害 (Denial of Service : DoS) 攻撃がある．DoS 攻撃とは，大量のパケットを対象ネットワークやサーバに送信することで，各種サービス提供を妨害する攻撃である．昨今では，インターネットを支えるネットワーク基盤の通信帯域やパケット処理能力が高まる一方で，DoS 攻撃の対象がよりサーバ側にシフトしてきた．これは，DoS 攻撃の予告あるいはその回避を示唆し，金銭を要求する脅迫や詐欺行為が増えていることからも把握できる [25].

サービス提供を DoS 攻撃から防御するには，DoS 攻撃がどのような影響を及ぼすのか理解する必要がある．インターネット通信が階層構造 (レイヤモデル) で設計されていることから，各階層における DoS 攻撃の影響について以下に述べる．

最初に，レイヤ 2 以下では，イーサネットフレームの増大による回線容量の圧迫が発生する．次に，レイヤ 3 では IP パケット数の増大によるルータへの過負荷が発生する．さらに，レイヤ 4 では，サーバ OS 内カーネルメモリ間のコピー頻度が増大し，TCP/UDP 通信に必要なソケットも浪費される．最後に，レイヤ 7 ではサーバ・アプリケーションの処理に係る負荷が増大する．

さまざまなDoS攻撃 [26][27] を階層モデルから見た場合，UDP FloodやPing Flood，Smurfなどは下位レイヤを圧迫する．SYN Floodは中間レイヤ，F5(リロード) 攻撃やHTTP-GET-Flood攻撃は上位レイヤに過負荷を生じさせる．前述のように，ネットワーク技術の進歩とサイバー犯罪への適応性から，近年では中間レイヤ以上を対象とするDoS攻撃が増加している．

一方で，DoS攻撃対策の研究には2つの潮流がある．DoS攻撃の検知と防御である．上記のレイヤモデルを参照すると，各レイヤに影響を及ぼす各攻撃が存在することから，検知手法は各レイヤごとに適用されることが望ましいと考える．しかし，大量のパケットやデータペイロードなど各レイヤのリソースを圧迫することから，防御手法は下位レイヤで実施されることが効率的であるといえる．

本論文では，DoS攻撃への耐性が高い防御手法を提案する．同手法を用いた透過型防御システム (B-DRIP : Bridge mode Directed Routing by Inspected Protocol) は，送信元IPアドレスを用いた経路制御 (ソースアドレスルーティング) を断続的に実施することで，DoS攻撃の勢いを寸断する．さらに当該防御システムは仮想ブリッジインタフェースによる透過型で設計されているため，運用管理しているネットワークやサーバシステム構成との親和性が高い．加えて，サービス妨害とサービス妨害攻撃の判別が困難 (false-positive) なF5攻撃への対応として，代理応答サーバをB-DRIPと組み合わせた運用管理手法を提示する．

以下に，DoS攻撃に対する関連研究，提案手法，実装について述べる．さらに実験の結果について，評価と考察を行う．

3.2 関連研究と問題点

DoS攻撃ではないが，slammerワームは，1台の感染機から376ByteのUDPデータを秒間あたり2,700パケット (pps) のレートで発生させるため，8Mbps近く回線を占有する [28]．slammerワームが発生した2003年当時では，数Mbpsの対外線を運用していた組織も多く，これらは通信不能に陥った．

上記における，感染機を抱えた組織内の対応は，感染機の場所を詳細に把握し，同機を組織内ネットワークから隔離することである．同様に，インターネットを介した大量パケット送信に基づくDoS攻撃でも，攻撃元を特定するためのトレース方式が研究されている [29][30]．これらは，攻撃元の端末群を攻撃元サイトへ通知したり，上位ISP(Internet Service Provider) へ遮断を依頼するなど，防御に関する研究といえる．しかし，各機関との広い連携を必要とするため，即応性や実現性が低い．

インターネットを介した DoS 攻撃が成立する理由は，サーバを抱えるホストが IP パケットの受信を制御できないことにある [32][31]．そのため，汎用的な DoS 攻撃防御手法として，ホストが受信パケットを制御する方式 [33][31] と，IP-based 方式が提案されている [32]．前者は，各ホストにパケット受信制御のメカニズムを導入する必要があるため，実現性が低く一般的とは言い難い．後者は，送信元 IP アドレスを基にホストやファイアウォールにフィルタリングを施すため，これまで大量パケットを発生させる DoS 攻撃への耐性が低いとされている [31][38]．

近年では，前節の階層モデルに照らして，上位レイヤに着目した防御手法が提案されている [34][35][36]．これらは，攻撃検知されたパケット群に対し，送信元 IP アドレスに応じて仮想マシンのリソース割り当ての増強や切替を行ったり，TCP レイヤでプロキシに迂回させるなど，上位レイヤで強制転送している．しかし，攻撃側は攻撃先が対抗措置を施すと，手法を変えて攻撃するとされている [25]．これは，下位レイヤに影響する攻撃に切り替えることを指しており，同一の階層で検知と防御を実施する上での構造的な問題である．したがって，各階層の負荷を一様に低減する総合的な防御手法が求められる．

一方で，これまでの防御手法では，その導入に際してホスト内のメカニズムの変更や，仮想マシンの制御などの負荷が避けられない．さらに，運用管理を継続する上で同手法による不具合が生じても，同手法が原因であると特定するための切り分けポイントの設定が困難である．

防御手法に関する問題とは別に，検知手法にも解決が困難な問題がある．異常なパケット数を閾値で検出する DoS 攻撃において，一般アクセスと DoS 攻撃によるアクセスの閾値設定が一意に決められない F5 リロード攻撃である [25]．これは，閾値によって DoS 攻撃と認定されたアクセスの結果を検証し，false-positive であったかをフィードバックさせることが困難なためである．一般には攻撃認定されたユーザによる，ページを閲覧できない旨の報告によって false-positive を把握するとされている [25]．しかし，ユーザには，ページの閲覧不可が DoS 攻撃認定に起因していることを判別できないため，運用管理で切り分けを実施する必要がある．

3.3 提案システム

筆者らはこれまでに，DoS 攻撃を回避するメカニズムを搭載した DoS 攻撃防御システムとして，Directed Routing by Inspected Protocol(以下，DRIP) を開発してきた [37][38]．DRIP は DoS 攻撃の送信元 IP アドレスを基に，当該 IP アドレスからのパケットを NULL デバイスなどに転送 (ソースアドレスルーティング) する．本方式によって，

ファイアウォールやルータ上の ACL(Access Control List) などによるフィルタリングよりも効率的に DoS 攻撃に対処することを示した．

しかし，DRIP はルータなどのレイヤ 3 ネットワーク機器と同様に，既存のネットワーク構成に導入するには経路制御を含むネットワーク構成を変更する必要がある．組織内外の対外接続部における構成変更は容易ではなく，DRIP の機器不良など障害が発生した場合，復旧対応の遅れが懸念される．さらに，DoS 攻撃はパケットの量的要因によって成立するため，DRIP では過剰に防御してしまう (false-positive) 事案に対応できない．

本論文における提案システムを，上記を改良した透過型として Bridge mode DRIP(以下，B-DRIP) とする．B-DRIP に必要な機能要件を以下に述べる．

- ソースアドレスルーティングで攻撃を逸らす
- 単位時間あたりのパケット数で判断する
- 仮想インタフェースによる透過型で構築する
- 代理応答サーバとの併用で false-positive に対応する

なお，本論文では，B-DRIP が機能要件を満たすこと，およびその性能を定量的に評価するために，検知・防御の対象となる DoS 攻撃として，下位層の攻撃であり防御が困難とされる SYN Flood 攻撃と，上位層の攻撃であり false-positive が発生しやすい F5 リロード攻撃を採用する．

上記の各要件について，以下に述べる．

3.3.1 ソースアドレスルーティング

これまでにも述べたように，送信元 IP アドレスに基づいて Next-hop を決定する方法として，筆者らはソースアドレスルーティングと呼称される経路制御方法を提案した [37][38]．本節では，ソースアドレスルーティングの挙動を明確にするために，後述の透過型を除いた従来の DRIP の要件として述べる．ソースアドレスルーティングは，送信元端末から宛先端末までの経路を一意に選択するソースルーティングとは異なる方式である．

ソースアドレスルーティングの挙動について，図 3.1 に示す．宛先端末 A に対して，攻撃端末と，一般アクセスを発生させる一般端末が接続する際，DoS 攻撃と認定された攻撃端末からのパケットは，同端末の送信元 IP アドレスに基づいて端末 X に転送される．本研究では，SYN Flood 攻撃の場合の転送先を NULL デバイスとし，F5 リロード攻撃の場合の転送先を，後述する代理応答サーバとする．筆者らは，以下の ipfw と zebra を用いてソースアドレスルーティングを実現している．

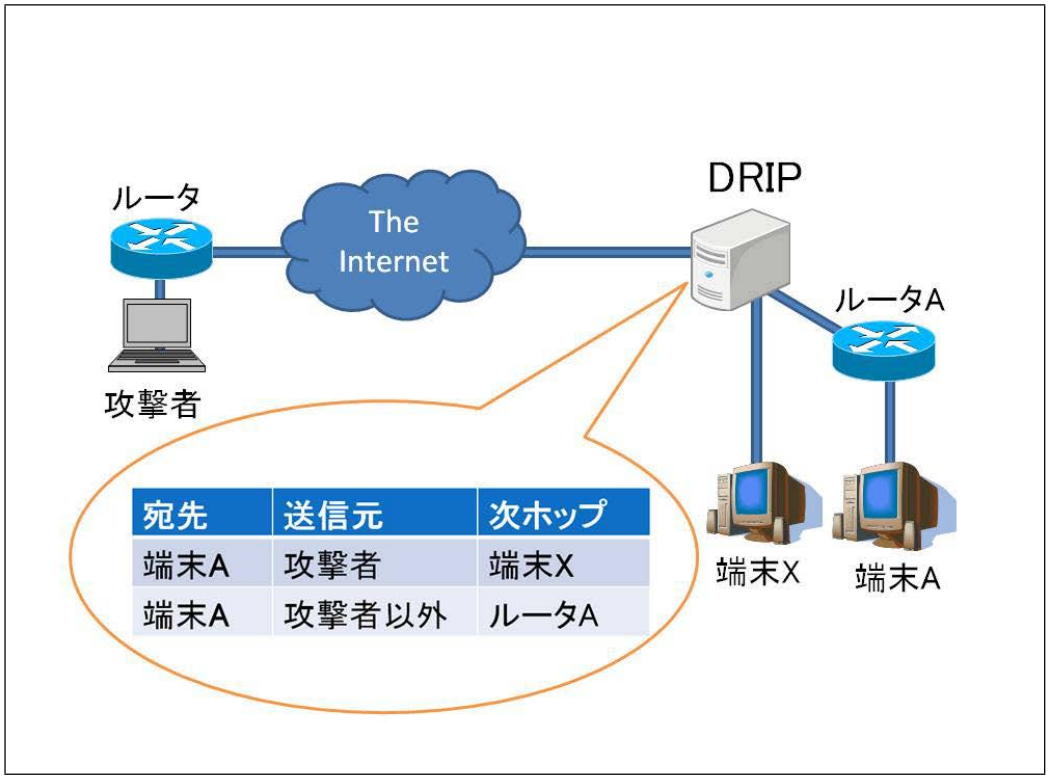


図 3.1　ソースアドレスルーティング

ipfw

FreeBSD には，インタフェースを出入りするパケットを監視し，あらかじめ指定された条件に合致するパケットをフィルタリングしたり転送する機能として，ipfw[39] がある．この転送機能を利用すると，送信元 IP アドレスをトリガにして転送することが可能となる．ipfw の転送ルール例を以下に示す．

ipfw add 10 fwd X.X.X.X all from Y.Y.Y.Y to any

上記のルールは，ルール番号 10 番において，送信元 IP アドレス Y.Y.Y.Y からのパケットをすべて宛先 IP アドレス X.X.X.X に転送する例である．ipfw の当該機能は，本来ネットワークインタフェースカード (NIC) を複数搭載したサーバ機において，サーバ機のデフォルト経路以外に，送信元 IP アドレスに応じて Next-hop を決めるために使用される．筆者らは，当該機能をネットワークの経路制御に流用した．

zebra

zebra[40] は，TCP/IP に基づく様々なルーティングプロトコルのマネージメントプログラムであり，経路情報をカーネルのルーティングテーブルに反映できる．zebra の経路記述の例を以下に示す．

ip route X.X.X.X/32 null0

上記の例では，宛先 IP アドレス X.X.X.X へのパケットを null デバイスに転送する．前節の ipfw では null デバイスが指定できない仕様となっている．したがって，DoS 攻撃と認定されたパケットは，当該パケット専用の宛先 IP アドレスである X.X.X.X に転送され，破棄される．

3.3.2 DoS 攻撃の検知手法

前述のように，階層構造における DoS 攻撃を検討した結果，防御を下位レイヤで実施することで耐性を高める一方，検知は各階層での攻撃手法に対応することが望ましい．本研究では，システムの有用性を示す実験に用いるため，多くの DoS 攻撃対策の研究と同じく，単位時間あたりのパケット数への閾値で検知する手法を採用する．

提案システムである B-DRIP は，DoS 攻撃への防御システムであると同時に，F5 攻撃など false-positive の回避が困難な攻撃を，B-DRIP による防御の中でフィードバックし，適時修正するメカニズムを搭載している．すなわち，上記の閾値を用いた一般的な検知手法において，false-positive の修正を実施する．

3.3.3 透過型の構築

本研究で実装した透過型である B-DRIP において, ソースアドレスルーティング等の基本的な動作は 3.3.1 節で述べた通りである．本節では, 先行研究の DRIP を透過的に投入できるようにする手法について述べる．

bridge カーネルオプション

FreeBSD 系 OS のカーネルオプションの 1 つに bridge カーネルオプションがある [41]. bridge カーネルオプションは, 2 つの NIC をもつ PC をブリッジのように動作させることが可能である．bridge カーネルオプションでは, 初めに仮想的なブリッジインタフェー

スを作成する（以下, 仮想ブリッジインタフェース）. 作成した仮想ブリッジインタフェースに, PC に搭載されている NIC を 2 つ登録することで, NIC 間でのイーサフレームのやり取りを可能にする.

以下に, 仮想ブリッジインタフェースを作成し, ブリッジさせる 2 つの NIC を登録するコマンド例を示す.

```
# ifconfig bridge create
# ifconfig eth0 up
# ifconfig eth1 up
# ifconfig bridge0 addm eth0 addm eth1 up
```

上記のようなコマンドを実行すれば, eth0 と eth1 の NIC 間でイーサフレームのやり取りが可能になる. 本研究では, bridge カーネルオプションを使用することで, B-DRIP を実装する.

3.3.4 DRIP の挙動

DRIP は基本的にルータとして機能する. 以下に, DRIP を用いた場合の攻撃回避の流れを示す.

1. 予め zebra が稼働している攻撃回避用 IP アドレス (DRIP 筐体で zebra が稼働していれば，同筐体内の仮想インタフェースのアドレスなど) を設定し, 攻撃回避用 IP アドレス宛てのパケットの送り先を, zebra を用いて/dev/null に設定しておく
2. 単位時間（1 秒間）あたりの閾値を設定し, 仮想インタフェースに流入するパケットをキャプチャする. 当該時間中に閾値を超えるパケットを送信したホストを攻撃者と判別する
3. 攻撃者と判別された IP アドレスからのパケットを, 攻撃回避用 IP アドレスに転送する ipfw ルールを発行する
4. 攻撃パケットが ipfw によって攻撃回避用 IP アドレスに転送される
5. 攻撃回避用 IP アドレスに転送されたパケットが, zebra によって/dev/null に送信される
6. 一定時間（5 分間）経過後, ipfw から当該ルールを削除する
7. 以降, 2. のパケットキャプチャ処理に戻る

なお, 攻撃者と認定された送信元 IP アドレスへの対応を 5 分間で解除することは, DoS 攻撃はパケットの量的要因によって成立する攻撃であるため，この勢いの寸断を狙うためである．世界中のプロバイダの中には，プライベートネットワークからの NAT(Network Address Translation) によってインターネットに接続している組織やプロキシ経由で接続する組織も存在する．したがって，上記のルールを永続的に適用することは，攻撃者以外の正規利用者も利用する IP アドレスからのアクセスを排除してしまうなど，運用上別の問題を発生させる可能性がある．

3.3.5 代理応答サーバ

本節では, F5 リロード攻撃に対して返信するために, ステートレスな TCP 代理応答を実行するサーバの構築手法について述べる．

概要

DRIP では, 攻撃者からのパケットを自在に転送することが可能である．先行研究の DRIP では, 攻撃パケットは/dev/null にルーティングさせることで, 攻撃パケットから各ホストや内部ネットワークを防御している．本研究では, F5 リロード攻撃のパケット (TCP の 80 番ポート宛て）は代理応答サーバに転送される．代理応答サーバでは, 受信した HTTP リクエストに対して, これが攻撃によるアクセスと判別された旨をテキストメッセージにし，HTTP リプライで返信する．ここで技術的な問題が発生する．B-DRIP も先行研究と同じく, ipfw を用いて実装しているが, fwd ルールで変更できるのは宛先 IP アドレスではなく, イーサフレームに含まれる宛先 MAC アドレスのみである．代理応答サーバが通常のサーバ機能をもっていた場合, 転送されてきた攻撃パケットは宛先 IP アドレスが自分のものではないため破棄されてしまう．よって代理応答サーバには, 宛先 IP アドレスが自分のものでなくとも受信し, 返信する機能が必要とされる．本機能を実装するために, ステートレスな TCP 代理応答という機能を考案し, 代理応答サーバに実装する．

ステートレスな TCP 代理応答

ステートレスな TCP 代理応答とは, 開放していないポート宛てに宛先 IP アドレスが異なる IP パケットが到着しても, 当該パケットに対して返信する機能である．図 3.2 と図 3.3 に, TCP 通信におけるセッションの確立と開放の図をそれぞれ示す．SYN パケットはセッションの確立要求時に, FIN パケットはセッション開放要求時に使用される．ACK

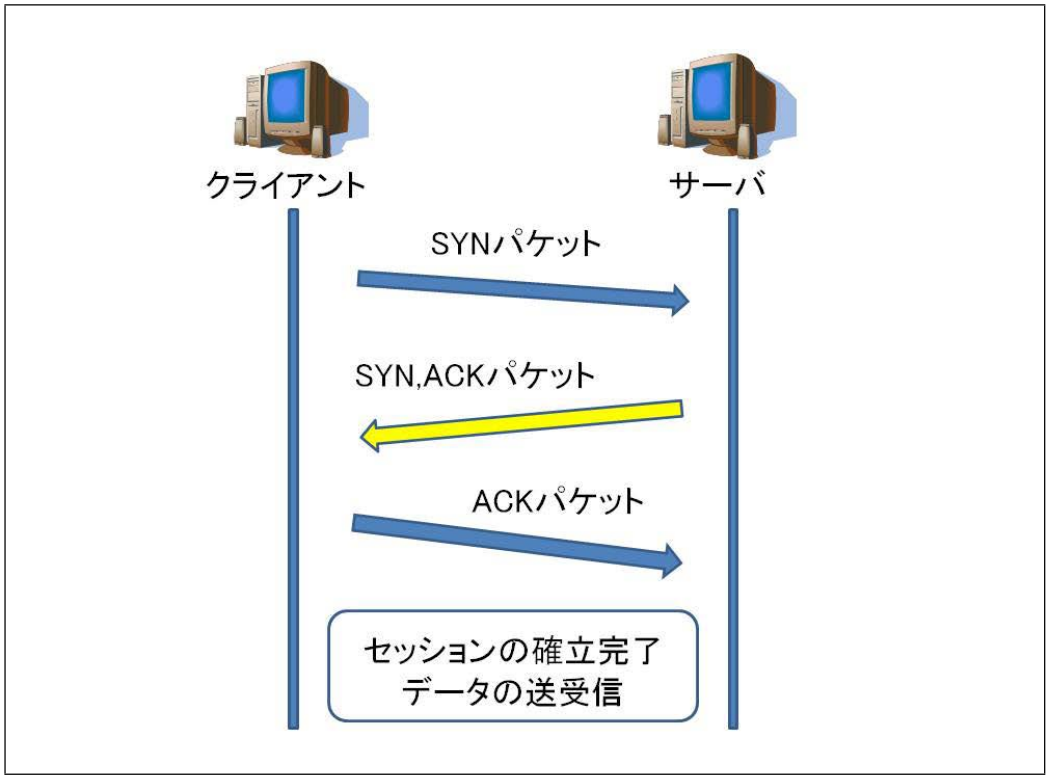


図 3.2　TCP セッション確立時の手順

パケットは, 各種要求に対する確認応答に用いられる．TCP 通信では, 通信の信頼性を保証するために図 3.2 と図 3.3 のようなパケットのやり取りが行われる．以下に，これらの制御パケットの到着パターンを適切に利用したステートレスな TCP 通信の手順について，HTTP 通信を例に挙げて示す.

1. 代理応答サーバは NIC に到着するパケットをキャプチャする
2. クライアントはサーバに対して, 80 番ポート向けに SYN パケットを送信する
3. 代理応答サーバは, キャプチャしたパケットが TCP の 80 番宛ての SYN パケットならば, クライアントが誤認して受け取るような SYN,ACK パケットを生成してクライアントに返信する
4. クライアントは受信した SYN,ACK パケットに対して ACK パケットを送信する
5. クライアントからサーバに対して, Web コンテンツを送るように HTTP リクエストが送られる
6. HTTP リクエストが代理応答サーバの NIC に到着した場合, その到着が，単位時間あたりの閾値を超えたアクセスであるため，F5 リロード攻撃と判別された旨を

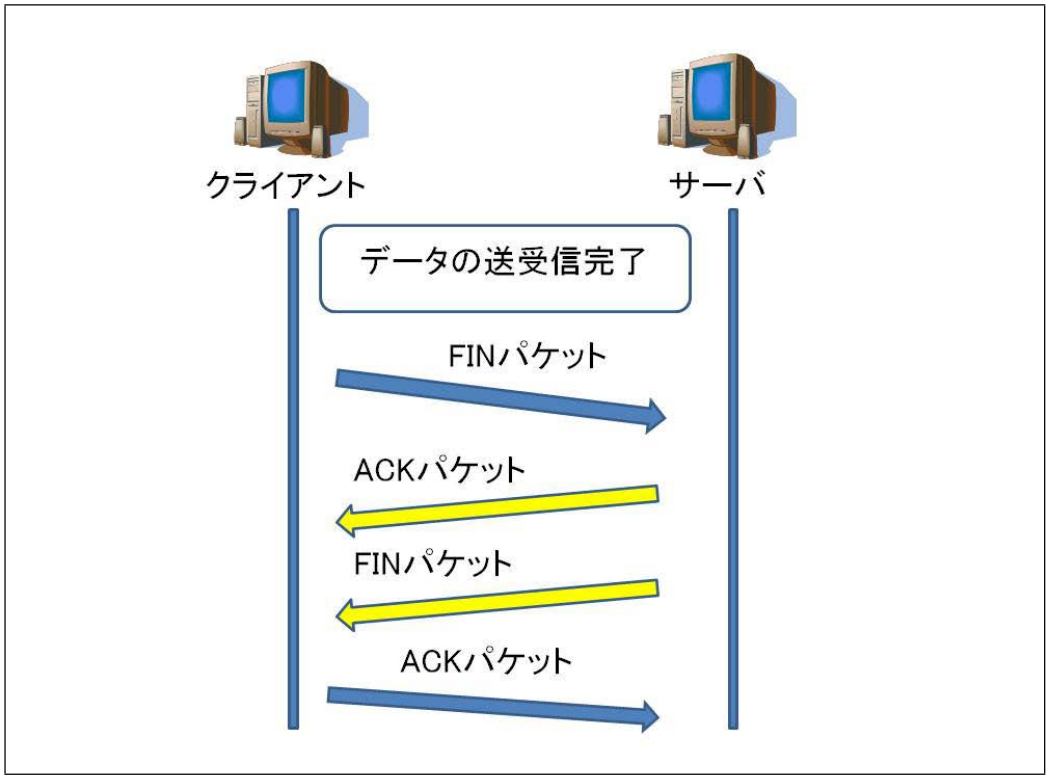


図 3.3　TCP セッション開放時の手順

テキストメッセージにし，HTTP リプライを作成してクライアントに返信する

7. 上記の 2 つ以外のパケットであれば処理は何もせず, 1. キャプチャ処理に戻る

代理応答サーバでは, セッション管理のためのリソース確保, 届いたパケットがセッションを確立した相手からのパケットかどうか等のチェックは一切行わない. 受信したパケットのチェックをしないため, 代理応答サーバはあらゆるパケットに対して返信が可能となる. さらにセッション管理をしないため, 負荷を軽減することが期待できる.

以上のような動作をするプログラムを, 高速な応答が見込まれる C 言語を用いて作成し, 代理応答サーバを実装する.

3.4　実装

本節では, 提案した手法を用いて実装したシステムについて述べる.

図 3.4 に提案システムの構成図を示す. ホスト宛てのパケットは, B-DRIP を通過してホストに届く. B-DRIP では, 管理者によって設定されたルールに基づいて, ソースアド

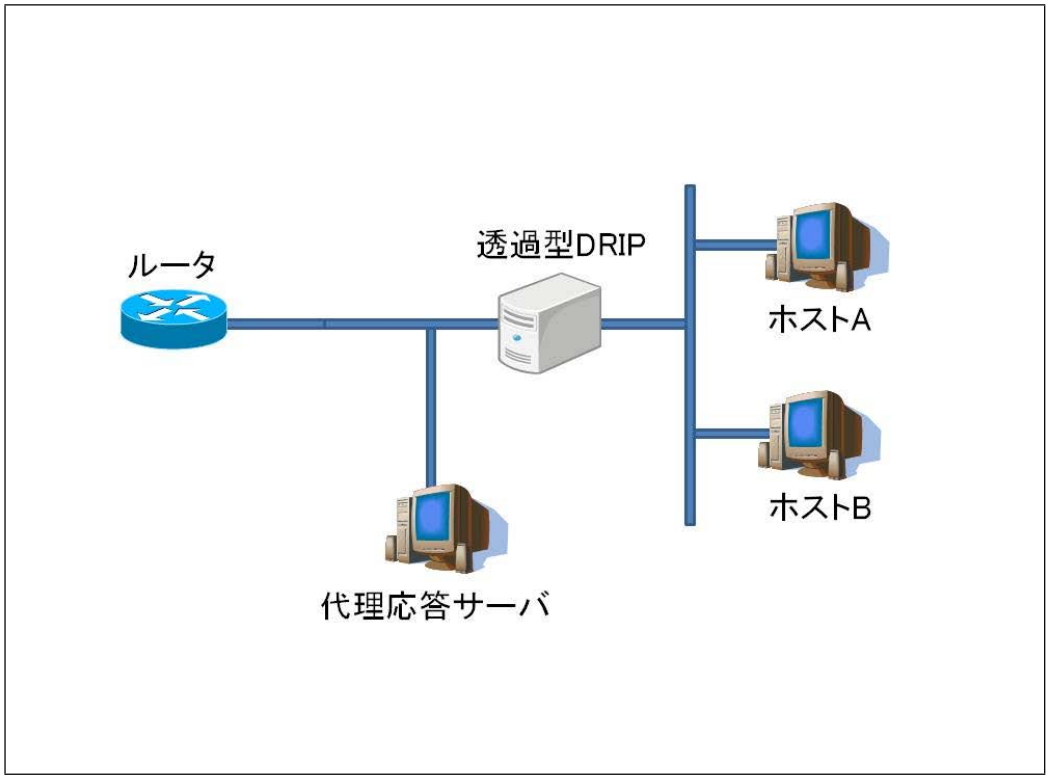


図 3.4　提案システムの構成図

レス毎に単位時間あたりのパケット数をカウントする．カウントした値が設定した閾値を上回っていた場合は，設定した対処法に基づいて処理される．対処時に，攻撃を代理応答サーバへ転送するという設定をすることにより，攻撃パケットを代理応答サーバが受信して返信をする．同様の処理が，ホストから外部ネットワーク宛てのパケットに対しても行われる．

3.4.1　B-DRIP

本節では，本研究で実装した B-DRIP のハードウェア構成，動作概要および期待する動作をさせるために，FreeBSD のカーネルソースコードを変更した点について述べる．

ハードウェア構成

実装したシステムのハードウェア構成を以下に示す．

- OS: FreeBSD 8.2-RELEASE

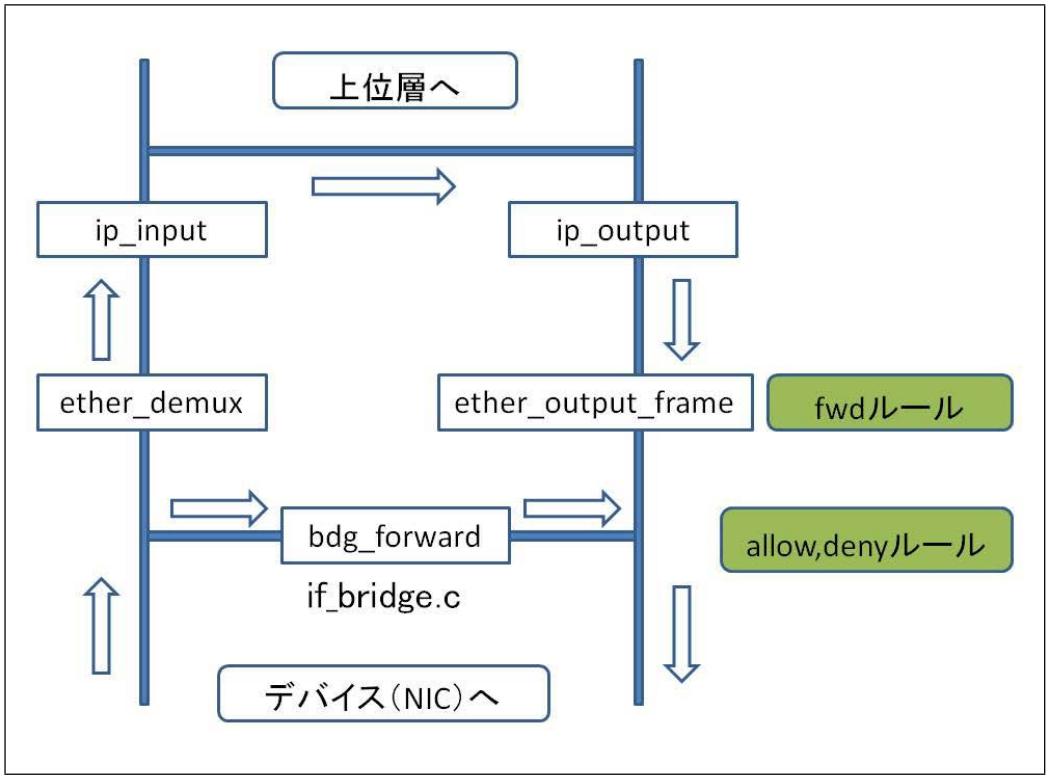


図 3.5　FreeBSD におけるパケットフロー

- CPU: Intel(R) Pentium(R) Dual CPU E2200 @2.20[GHz]
- メモリ: 2[GB]
- NIC1: Broadcom Gigabit Ethernet Controller
- NIC2: Intel Gigabit CT Desktop Adapter EXPI9301CT

カーネルソースコードの変更

B-DRIP の基本的な部分は, ipfw の fwd ルールと bridge カーネルオプションで成り立っている．しかし, ipfw の fwd ルールは仮想ブリッジインタフェースで取り扱う, イーサフレームに対応していないという問題が発生した．図 3.5 に, FreeBSD におけるパケットフローと ipfw の主なルールが, どのプロトコルスタック上で動作可能であるかを示す. bridge カーネルオプションを使用してブリッジングされるパケットは, bdg_forward を経由して処理される．この部分では, ipfw の allow や deny といったルールを適用できる．しかし, bdg_forward の部分では，fwd ルールは適用できない.

そのため B-DRIP では, bdg_forward を通過する（ブリッジングされる）パケット

にも fwd ルールが適用されるようにカーネルソースコードを変更した．具体的には，bdg_forward 部分を構成している if_bridge.c ファイルを変更した．変更により，bdg_forward から ether_output_frame を経由することによって，ブリッジングされるパケットにも fwd ルールが適用されるようになった．しかし，変更できるのはフレーム中に含まれる宛先 MAC アドレスのみであり，パケット中の宛先 IP アドレスは変更前のままである．

B-DRIP の動作概要

B-DRIP の動作や，当該アクセスを攻撃と判別した場合の対処法について述べる．

■基本動作　透過型である B-DRIP では，先行研究の DRIP と同じように，当該アクセスの攻撃判定を単位時間あたりのパケット数を用いて判別する．本研究では単位時間を 1 秒間とし，1 秒間で設定した閾値を超えるアクセスをした者を攻撃者と判別する．攻撃者は 5 分間だけ攻撃者リストに情報が載るが，5 分経過したら一旦情報が削除される．攻撃者リストに載っている者は，設定されたルールによって対処される．

攻撃判定は C 言語で作成したプログラムを用いてパケットをキャプチャし，ルールファイルに当てはまるパケットを 1 秒間カウントして，カウントした数と閾値を比較する．閾値を超えている場合は，ルールファイルに記された対処が実行される．

■ルールファイル　ルールファイルには防御対象となるホストの IP アドレス，ポート番号，閾値および攻撃への対処法を設定する．図 3.6 にルールファイルの書式例を示す．

■攻撃への対処法　B-DRIP では，攻撃者であると判別された者に対して 3 つの対処法を用意している．

- null: 先行研究の DRIP と同じように攻撃パケットを/dev/null に転送する
- fwd: 攻撃パケットを代理応答サーバに転送する
- alert: ipfw 等を用いた対処はせず，ターミナル上と syslog にアラートを出す

3.4.2　代理応答サーバ

本節では，F5 リロード攻撃と判別したアクセスに返信する代理応答サーバの構成や動作トリガーについて述べる．

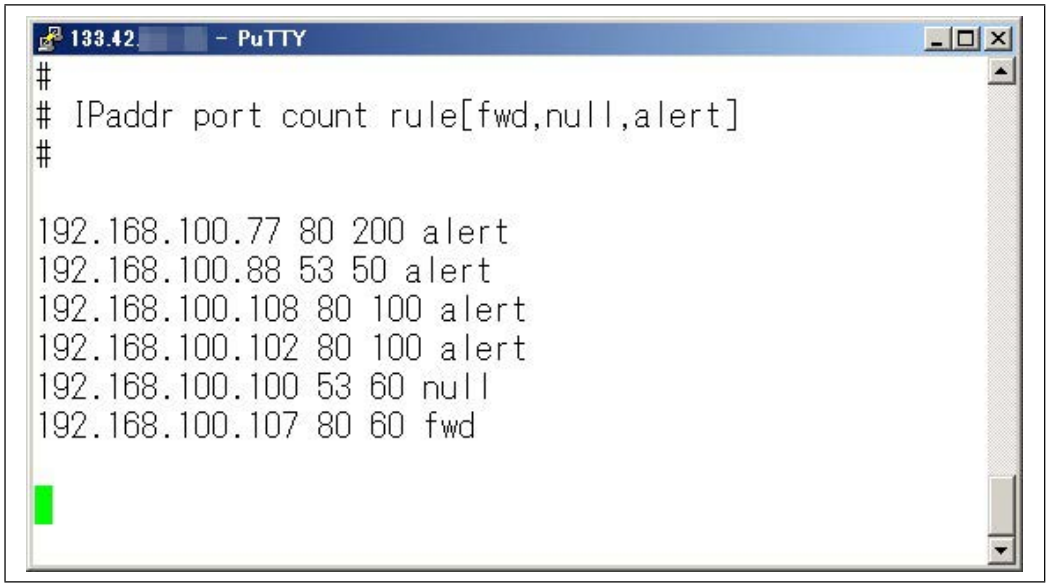


図 3.6　ルールファイル例

ハードウェア構成

実装した代理応答サーバのハードウェア構成を以下に示す.

- OS: Vine Linux 5.2
- CPU: Intel(R) Pentium(R) Dual CPU E2200 @2.20[GHz]
- メモリ: 2[GB]

代理応答サーバの動作概要

代理応答サーバは, TCP ポートの開放や宛先 IP アドレスに関わらずに返信する機能を有する. 図 3.7 に代理応答サーバの動作トリガを示す. 代理応答サーバの目的は, F5 リロード攻撃者に対して, 同攻撃だと判別された旨の警告コンテンツを表示させることである. コンテンツを表示させるには, TCP セッションを確立した後に, 攻撃者に対して表示させたいコンテンツを HTTP リプライで返信する必要がある. 図 3.7 からわかるように, 代理応答サーバは 2 種類のパケットを偽装して送信すれば, 攻撃者に対して警告コンテンツを表示させることができる. 具体的な代理応答サーバの動作手順を以下に示す.

■SYN, ACK パケットの送信手順

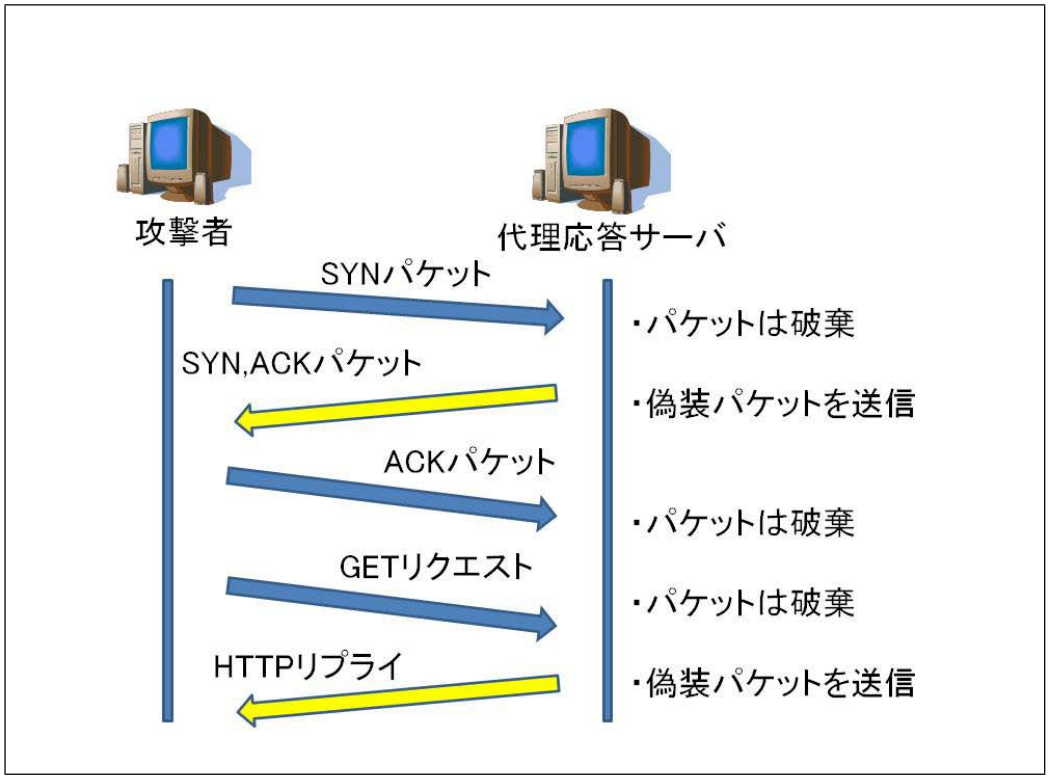


図 3.7　代理応答サーバの動作トリガー

1. 代理応答サーバの NIC に到着するパケットをキャプチャする
2. TCP の 80 番ポート宛てに SYN パケットが到着すると，偽装パケットの作成を開始する
3. 宛先 IP アドレスと送信元 IP アドレスを入れ替える
4. TCP ヘッダの宛先ポートと送信元ポートを入れ替える
5. TCP の 3way-handshake 時と同様に，受信したパケットの TCP シーケンス番号に 1 を足したものを確認応答番号とする
6. 送信するパケットに任意のシーケンス番号をつける
7. TCP ヘッダの SYN, ACK フラグビットをたてる
8. 各種チェックサムを計算した後にパケットを送信する
9. 1. のパケットキャプチャ処理に戻る

■警告コンテンツの送信手順

1. 代理応答サーバの NIC に到着するパケットをキャプチャする

2. TCP の 80 番ポート宛てに PSH, ACK パケットかつ GET リクエストが到着すると, 偽装パケットの作成を開始する
3. 宛先 IP アドレスと送信元 IP アドレスを入れ替える
4. 用意した HTTP コンテンツのデータサイズ分を考慮した IP ヘッダ長を登録する
5. TCP ヘッダの宛先ポートと送信元ポートを入れ替える
6. 受信したパケットの確認応答番号をシーケンス番号へ登録する
7. TCP のデータ受信時と同様に，受信したパケットのシーケンス番号に TCP 受信データ長を加算して確認応答番号へ登録する
8. 用意した警告コンテンツのデータを TCP データへコピーする
9. 各種チェックサムを計算した後にパケットを送信する
10. 1. のパケットキャプチャの処理に戻る

3.4.3 管理運用フレームワーク

代理応答サーバは, 攻撃検出時の false positive を本システムの運用にフィードバックするため, 警告コンテンツを返信する．システムに問題や改善点があった場合に, 問題の原因究明や改善策を見つけ出す．見つけ出した改善策等をシステムに反映させて運用を続ける．本サイクルを繰り返すことを, “管理運用フレームワーク”と呼ぶ．本システムでも, Web サーバのコンテンツ量が増えれば, 予め設定していた閾値を上回ってしまい, 正規のユーザを攻撃者だと判別してしまう．このような false positive を早期に発見するためのフィードバックとして警告コンテンツの返信を行う．正規のユーザから, 「警告コンテンツの表示があった」と管理者に対して問い合わせがあった場合は, 閾値の見直し等を行う．事例によっては, DDoS 攻撃の踏み台にされた端末を発見することも可能となる．警告コンテンツにより, 様々な情報を管理運用フレームワークへフィードバックすることができる．

3.5 動作実験と結果

本節では, 実装した提案システムの動作実験および結果について述べる．

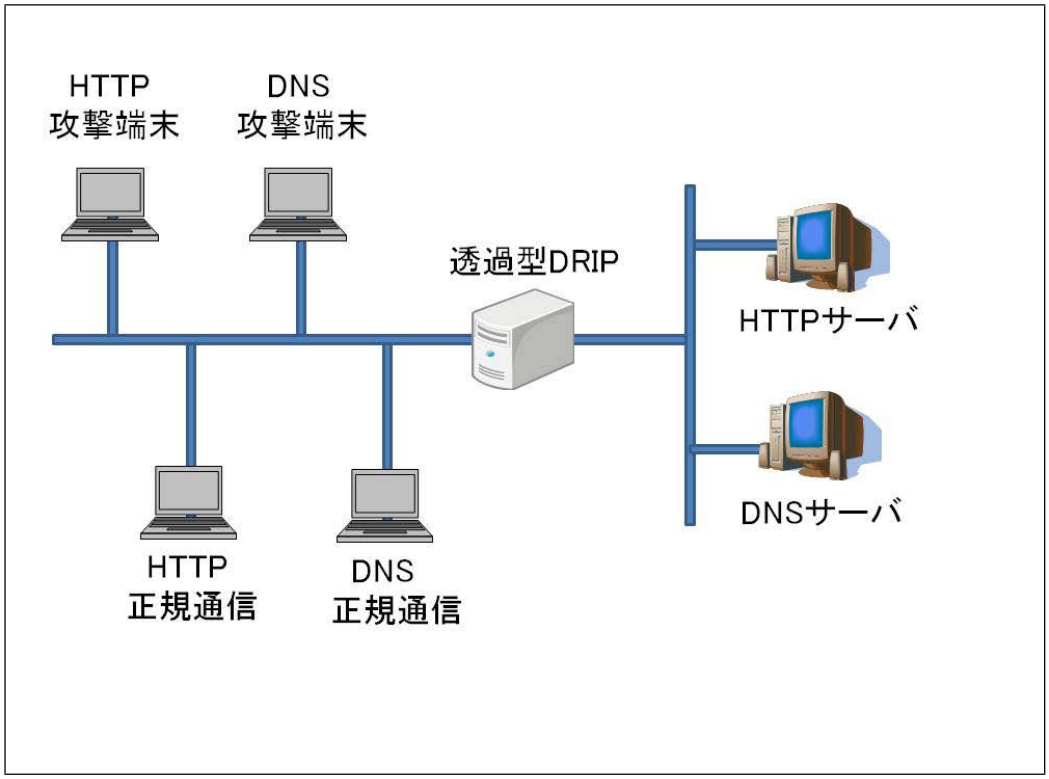


図 3.8　B-DRIP の負荷実験環境

3.5.1　B-DRIP

透過型である B-DRIP では DoS 攻撃に対する負荷実験と, 実際に和歌山大学の学内ネットワークに導入する運用実験を実施した.

3.5.2　負荷実験

本実験は, 設置している DNS サーバ 1 台と HTTP サーバ 1 台に, それぞれ DoS 攻撃が実行されたケースを想定している. 攻撃には UDP Flood 攻撃と SYN Flood 攻撃を使用する. 攻撃を実行しても, B-DRIP が設定されたルールに基づき, 正確に攻撃対処ができることを検証した. さらに, F5 リロード攻撃の実行中でも正規通信には影響がないことも検証した.

実験環境

実験は攻撃の特性を考慮し, クローズドネットワークで 60 秒間実施した. DNS サーバに対して UDP Flood 攻撃を実行する端末を 1 台, HTTP サーバに対して SYN Flood 攻撃を実行する端末を 1 台用意した. また, 各サーバに正規通信を想定したトラヒックを発生させる端末を 2 台用意した. 図 3.8 に実験環境を示す. 実験の流れは以下の通りである. なお, 本論文では 1 秒間あたりのパケット数を示す単位を [pps] と表記する[*1].

1. 実験開始前に, B-DRIP の攻撃検出の閾値を DNS サーバは 6000[pps], HTTP サーバは 8000[pps] と設定する. これは, 後述する正規通信の想定トラヒックを超える閾値として採用した.
2. 攻撃検出時の対処法はどちらも null デバイスへの転送とする
3. 実験開始時に正規通信として, DNS サーバに対して DNS 正規端末から 4000[pps], HTTP サーバに対して HTTP 正規端末から 6000[pps] のトラヒックを発生させる. これら正規通信のトラヒックは, 和歌山大学の対外接続部における 5 分間トラヒックの平均最大値が 10,000-12,000pps であることから採用した. 正規通信トラヒックの発生と同時に DNS サーバと HTTP サーバの到達パケット数のカウントと, DNS サーバ, HTTP サーバおよび B-DRIP の CPU 負荷の計測を開始する.
4. 実験開始から約 20 秒後に, DNS サーバの 53 番ポートに対して秒間 20000[pps] の UDP Flood 攻撃, HTTP サーバの 80 番ポートに対して 50000[pps] の SYN Flood 攻撃のトラヒックを発生させる. 具体的には, 各サーバへの攻撃コマンドを攻撃端末上で手動にて実行するので, 20 秒の前後の時点で当該トラヒック量に達すると考えられる.
5. 実験開始 40 秒後, B-DRIP による攻撃対処を開始する. 高負荷時のグラフを見やすくするため, 手動による B-DRIP のプロセス実行を目測 40 秒の時点で行う.

実験結果

図 3.9 は実験時の各サーバの CPU 負荷率の推移を示すグラフである. また, 図 3.10 に各サーバへの単位時間あたりのパケット到達数の推移を表すグラフを示す. 図 3.9 を見ると, 2 つのサーバの CPU 負荷は, 目測による時間計測と手動による実行なので厳密な時刻と前後するが, 実験開始から約 20 秒後急激に増加していることがわかる. 実験開始から

[*1] Packet Per Second: パケット毎秒を表す

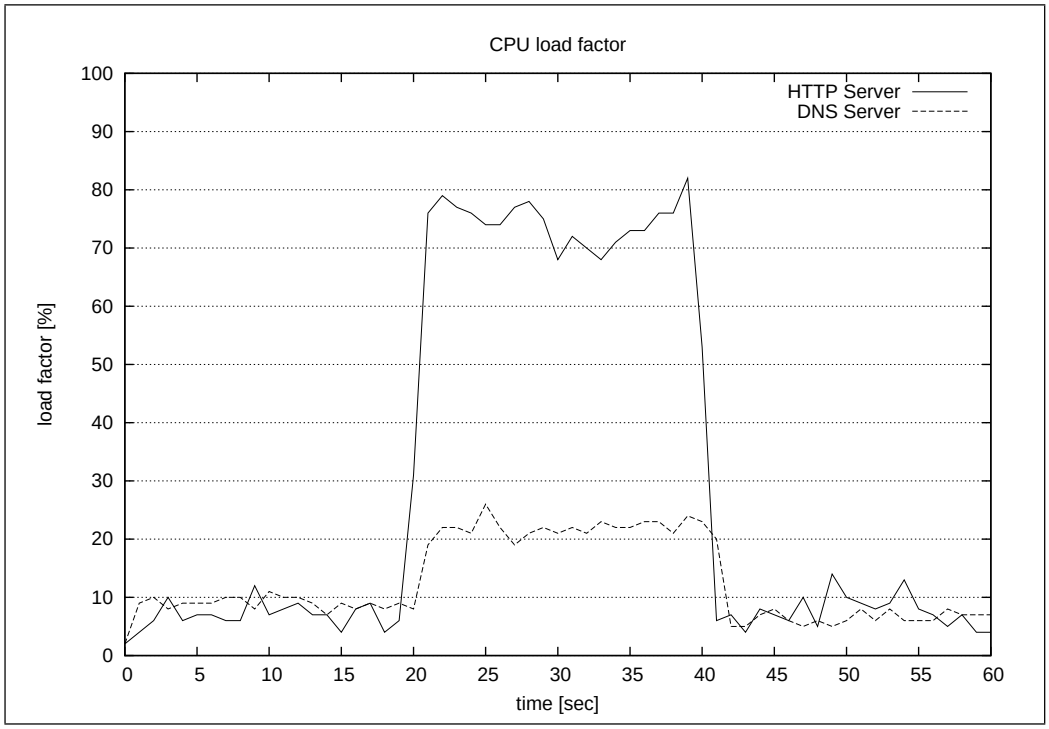


図 3.9　各サーバの CPU 負荷率

約 40 秒後では, CPU 負荷が急激に減少していることが確認できる．これは B-DRIP によって攻撃の検出および対処が適切に実行されたためである．図 3.10 のパケット推移を見ても, 攻撃対処後は攻撃開始前の正規通信用のパケット数と同程度であることがわかる．よって実装した B-DRIP は, 攻撃パケットと正規通信パケットを正確に判別していることを確認できた．

図 3.11 に B-DRIP の CPU 負荷率のグラフを示す．実験開始 20 秒後は, 攻撃パケットを大量にブリッジングしているため負荷が急増していることがわかる．実験開始 40 秒後では, ソースアドレスルーティングを開始するが, ソースアドレスルーティングにより CPU 負荷は 30% しか増加していないことがわかる．

実運用実験

3.5.2 節の実験結果から, 実装した B-DRIP が実際の組織内ネットワークで運用可能であると判断した．したがって, 和歌山大学のネットワーク管理者の許可を得た後, 学内ネットワークでの実運用実験を開始した．研究目的で述べた通り, ネットワークやサーバ構成等の変更は一切せずに, ケーブルの差し替えのみの作業で導入することができた．

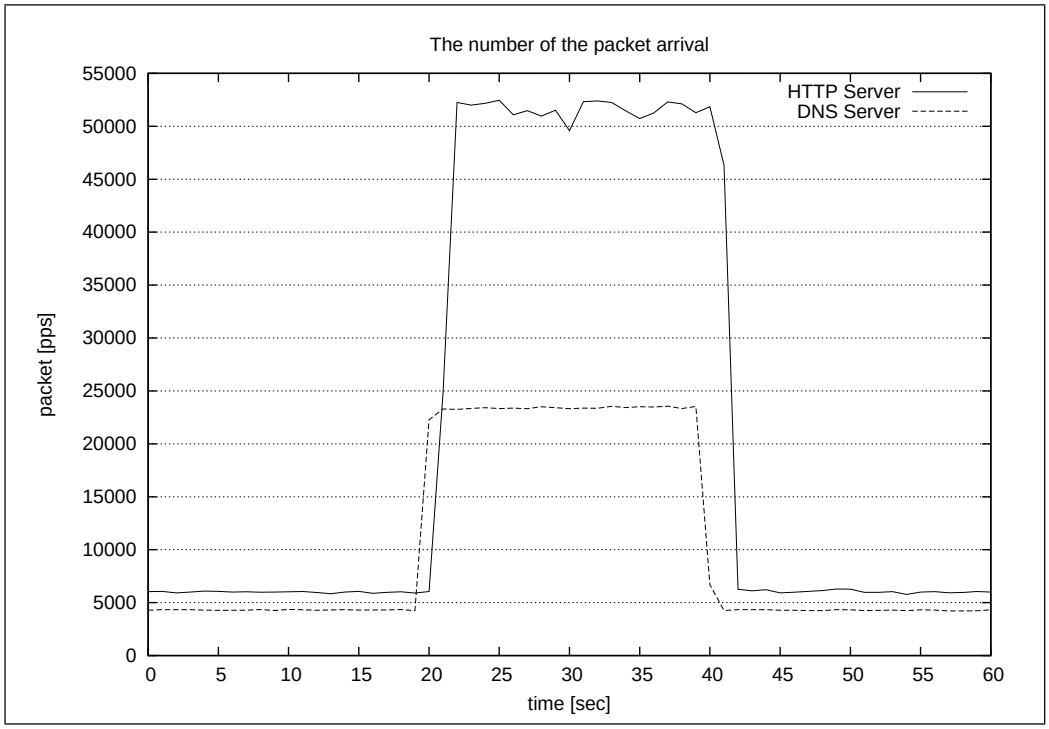


図 3.10　各サーバへのパケット到達数

3.5.3　TCP 代理応答サーバ

代理応答サーバの動作実験では, 機能証明と耐久性テストを実施した.

機能証明

実装した代理応答サーバが正しく機能しているかどうかを証明するために, 機能証明の実験を実施した. 図 3.12 に機能証明の実験環境とパケットのキャプチャポイントを示す. 端末は F5 リロード攻撃端末と正規ユーザ端末の 2 台である. F5 リロード攻撃端末は, Web サーバにアクセスする際に F5 キーを押し続けることで同攻撃を実行する. Web サーバに関して, 和歌山大学の Web トップページ（`http://www.wakayama-u.ac.jp/`）と同程度のパケットの送受信が行われるページを用意した. また, 図 3.13 に代理応答サーバが返信する警告コンテンツを示す.

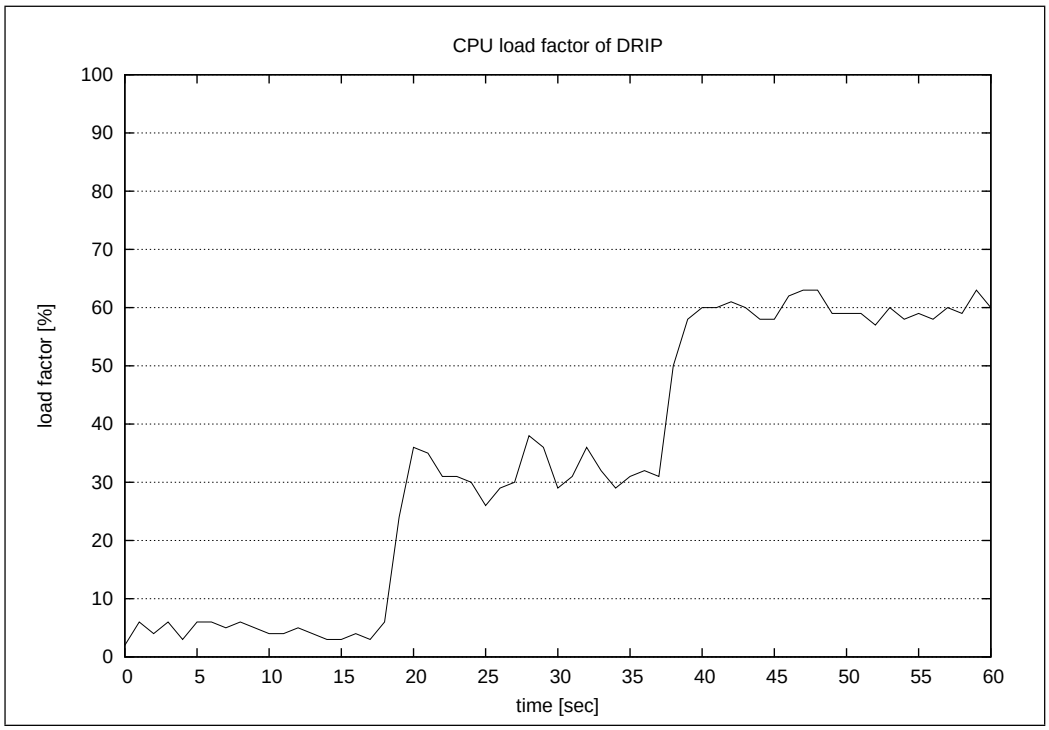


図 3.11　B-DRIP の CPU 負荷率

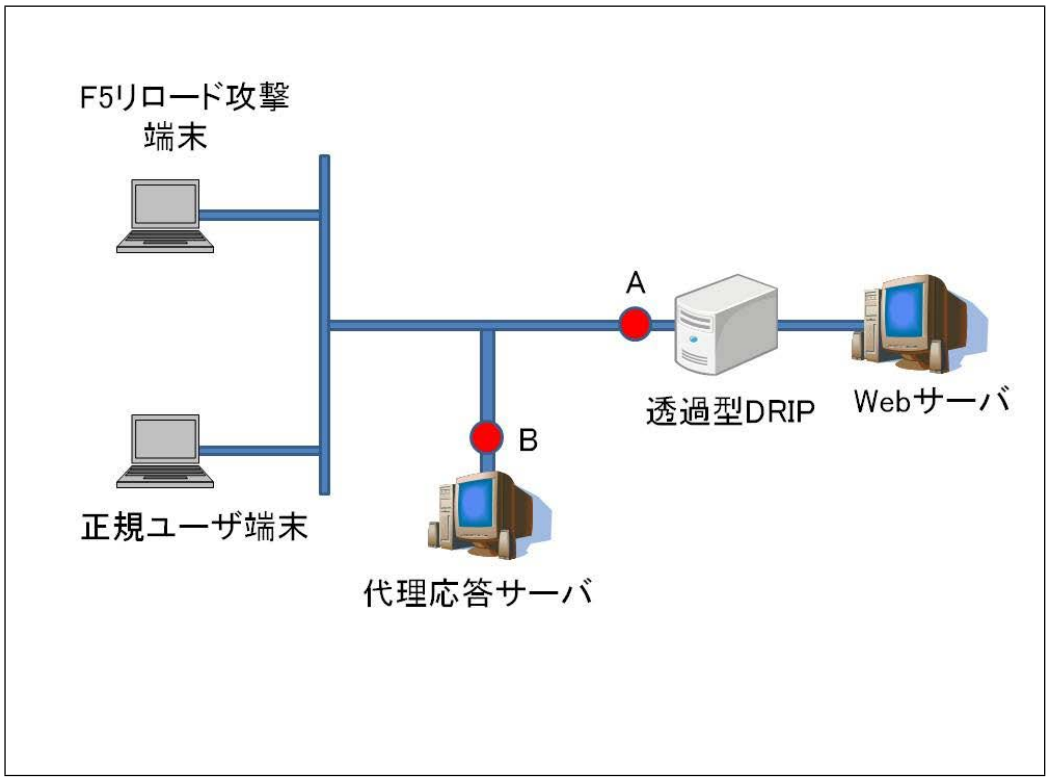


図 3.12　実験環境とキャプチャポイント

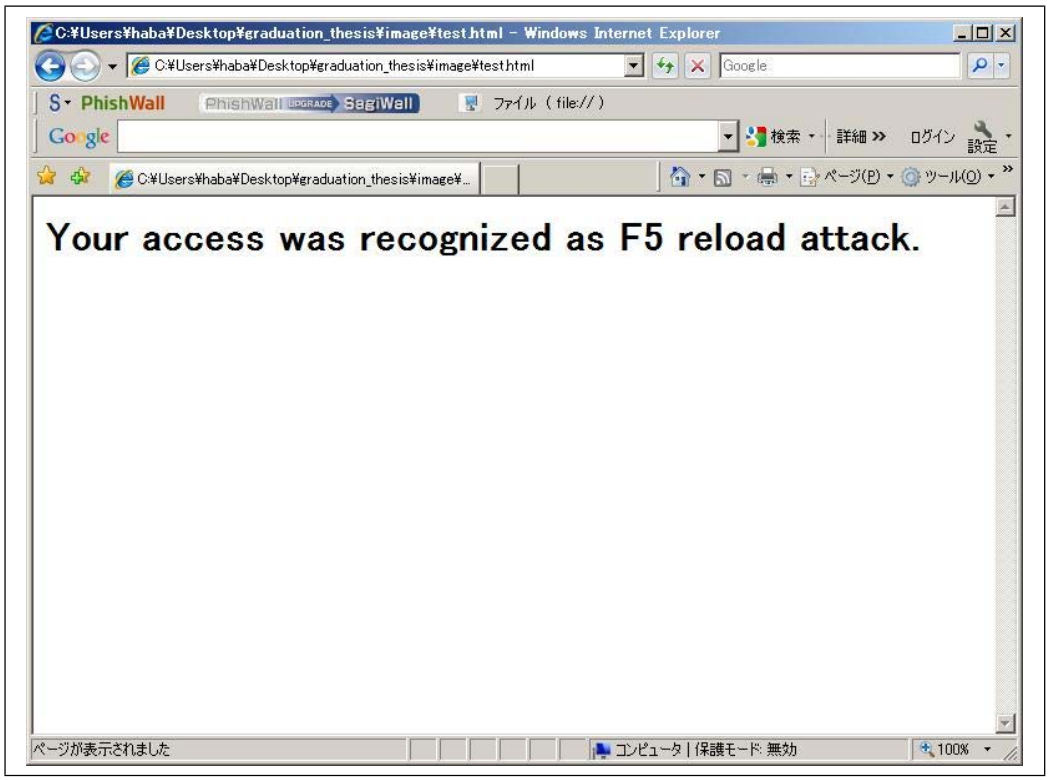


図 3.13　警告コンテンツ

各端末から, Web サーバや代理応答サーバにアクセスした際に発生する送出パケット数は表 3.1 のようになっている．Web サーバには，和歌山大学のトップページコンテンツを掲載しており，代理応答サーバへのアクセスには図 3.13 が返信される．

すなわち，正規ユーザ端末からのアクセスでは，それぞれにおよそ 400pps と 6pps が観測され，F5 リロード攻撃で各コンテンツにアクセスすると，およそ 1100pps と 25pps が観測される．これは，正規コンテンツよりも警告コンテンツを返信する方が，大幅にパケット数を低減できることを示す.

表 3.1　各端末でアクセスした場合の平均送出パケット数

	Web サーバ	代理応答サーバ
正規ユーザ端末	400[pps]	6[pps]
F5 リロード攻撃端末	1100[pps]	25[pps]

攻撃端末からのアクセスは, 代理応答サーバへ転送されるように B-DRIP の閾値を 800[pps] と設定した．代理応答サーバの機能証明としては, 図 3.12 のポイント A での攻撃端末からの流入パケット数と, ポイント B での攻撃端末向けの流出パケット数をカウントする．ポイント A とポイント B でのキャプチャ量が同程度で, 攻撃端末のブラウザ上に図 3.13 の警告コンテンツが表示されれば代理応答サーバは正しく動作していることになる.

実験結果

実験は 10 秒間実施して, 攻撃端末上に警告コンテンツが表示されることを確認した．各ポイントでの経過秒数毎のパケットキャプチャ結果を表 3.2 に示す.

表 3.2　各ポイントでのパケットキャプチャ数 [pps]

	1 秒	2 秒	3 秒	4 秒	5 秒	6 秒	7 秒	8 秒	9 秒	10 秒
ポイント A	30	30	29	30	31	28	31	29	30	30
ポイント B	30	30	29	30	31	28	31	29	30	30

表 3.2 の結果から，ポイント A とポイント B でのキャプチャパケット数の相違は見られない．すなわち，B-DRIP で転送された GET リクエストの秒間パケット数と代理応答サーバから返信される HTTP リプライの秒間パケット数は一致する．HTTP リプライには軽いテキストデータによる警告コンテンツが含まれているため，1 個のパケットのみ返

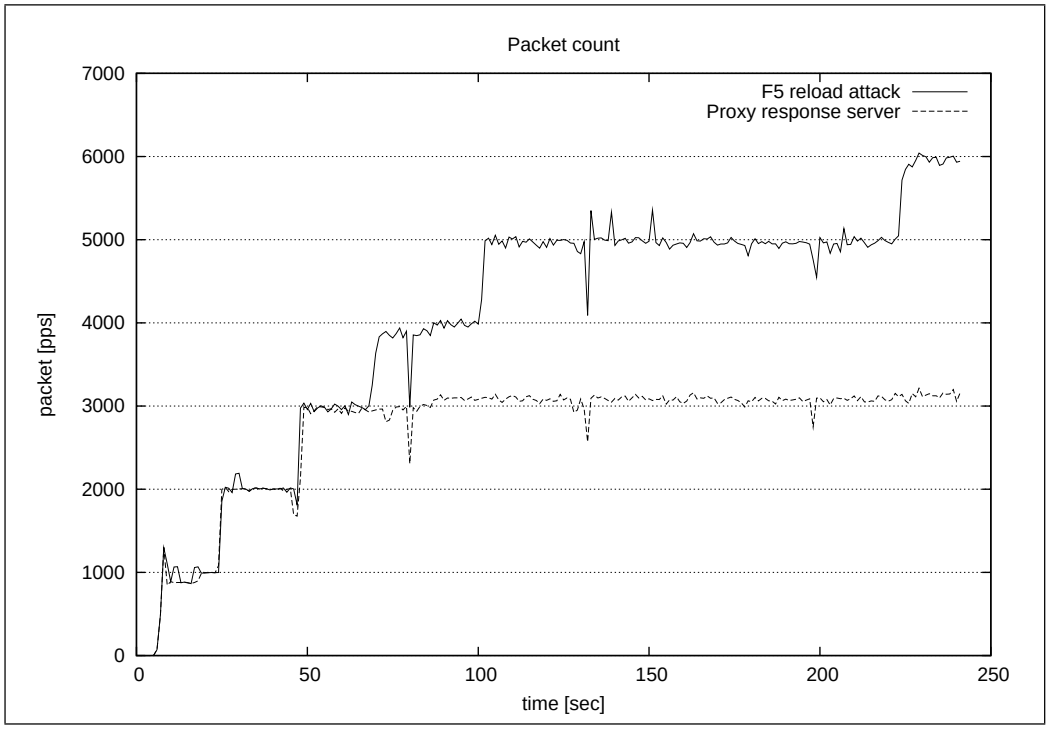


図 3.14　攻撃パケット数と返信パケット数

信される．よって, 代理応答サーバは正しく動作していることを確認した.

3.5.4　耐久テスト

耐久テストでは, 実装した代理応答サーバがどの程度の F5 リロード攻撃に耐えることが可能かを検証した．代理応答サーバの動作は, SYN, ACK パケットの返信か警告コンテンツの返信のどちらかである．今回の耐久テストでは, SYN Flood 攻撃を用いて耐久性のテストを実施した．これは，F5 リロード攻撃よりも SYN Flood 攻撃の方が大量のパケットを発生させるためであり，同時に，代理応答サーバは SYN パケットには SYN/ACK パケットを，GET リクエストには警告コンテンツのリプライを返信するので，耐久テストとして前者の組み合わせを採用した．徐々に SYN Flood 攻撃の量を増加させて, 同攻撃に対する代理応答サーバの返信パケット数がどのように変化するかを検証した．実験環境とパケットキャプチャポイントは, 図 3.12 と同じである.

実験結果

図 3.14 に攻撃端末からの流出パケット数と, 代理応答サーバの返信パケット数のグラフを示す. 図 3.14 を見てみると 3000[pps] までは, 攻撃端末と代理応答サーバのパケット数の相違はほぼ見られない. しかし, 攻撃パケットが 4000[pps] 以上になっても代理応答サーバの返信パケット数は増えていないことがわかる. よって, 実装した代理応答サーバは 3000[pps] までは完全に返信可能な機能をもつことが判明した. 部分的にパケット数が下がっている箇所がある. これは, B-DRIP と転送先である代理応答サーバを接続しているネットワーク機器のストア・アンド・フォワード機能による瞬間的な落ち込みであると推測され, 実験上の影響は無いと考える.

3.5.5 B-DRIP と TCP 代理応答サーバの動作実験

本節では, B-DRIP と代理応答サーバを用いて, より実運用に近い形での動作実験を実施した. 当該実験は, PC を複数台用意して実際に Web サーバに F5 リロード攻撃をするものである. PC を 10 台用意して実験を行い, それぞれ同時に F5 リロード攻撃を開始する. その状態で各 PC に警告コンテンツが表示されるかを確認し, 図 3.12 と同じポイントで流入パケット数と流出パケット数をキャプチャした.

図 3.15 に F5 リロード攻撃端末からの流入パケット数と, 代理応答サーバからの流出パケット数のグラフを示す. 実験開始 10 秒後で, 急激に F5 リロード攻撃のパケット数が増えていることがわかる. しかし, 直後から B-DRIP により攻撃と判別されるためパケットはすべて代理応答サーバへ転送される. そのためパケット数は, 250[pps] 程度で安定している. 表 3.1 より, F5 リロード攻撃端末が代理応答サーバへアクセスした場合は, 平均 25[pps] 程度のトラヒックが発生する. 今回の実験では, 10 台の PC を用いているため, 攻撃パケットは約 250[pps] で安定していると推察する. 実験では, すべての攻撃端末上に警告コンテンツが表示されることを確認した. また, 攻撃中であっても正規ユーザは正常に Web ページを閲覧できることも確認した.

3.6 考察

本節では, 3.5 節での実験結果を踏まえて考察を述べる.

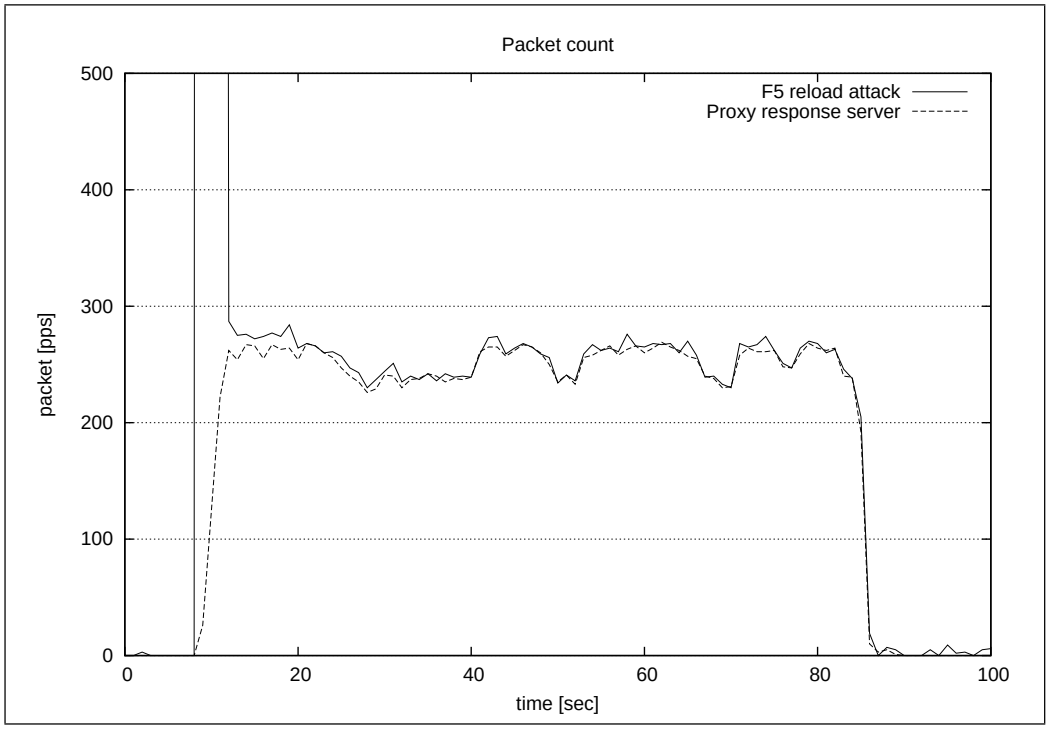


図 3.15　流出パケット数と流入パケット数

3.6.1　B-DRIP

ソースアドレスルーティングによる DRIP というシステム自体は先行研究で提案されてきたものである．本研究では, 導入時にネットワークやサーバ構成の変更をしなければならない DRIP を, 透過的に導入することを目的とした．bridge カーネルオプションを用いることで DRIP をルータからブリッジのように動作するよう実装した．先行研究では, 構成変更の問題から実運用評価を行うことができなかったが, 本研究の B-DRIP はケーブルの差し替えのみで学内ネットワークへの導入が可能であった．ネットワーク上で B-DRIP の設置位置を変更する場合も, ケーブルの差し替えのみであらゆる場所に設置が可能となる．

以上のことから, B-DRIP は様々なネットワークへ柔軟に導入可能なシステムであるといえる．ネットワーク運用の観点から見ても, ネットワークやサーバ構成の変更という煩雑な作業が発生しないことは大きなメリットである．

さらに, ipfw のルール上にダミールールが存在する場合でも, B-DRIP は正しく動作することを確認した．3.5.3 節の図 3.12 と同じ環境下で， ダミールールが 1000 個までの実

験を実施したが, 実運用上, これ以上のルールが増えることは考えにくい. よって, ipfw のルール数による負荷はほぼ無いものと推察できる.

3.6.2 TCP 代理応答サーバ

代理応答サーバは, false positive のフィードバックを目的として警告コンテンツを返信する. 代理応答サーバに通常のサーバ機能をもたせても, 転送されるパケットの宛先 IP アドレスが違うため, パケットを受信しない問題があった. 問題を解決すると同時に, DoS 攻撃への耐性を高めるためにステートレスな TCP 代理応答を提案した. 3.5 節の実験から, B-DRIP と代理応答サーバを組み合わせて, F5 リロード攻撃者に警告コンテンツの返信をすることができた. 仮に, 悪意のない正規ユーザに警告コンテンツが表示されたとしても, 表示された旨を管理者に伝えることで, 管理運用フレームワークに役立てることが可能となる. 攻撃者と認定された原因の究明や, 閾値の再設定をするという運用へ繋げることができる.

また, 図 3.14 の結果より, 現在の代理応答サーバは約 3000[pps] の返信能力しかもたないが, B-DRIP で攻撃を判別できれば, 攻撃自体は防御対象のホストへ到達することはない. 警告コンテンツの返信も, すべての攻撃に対して完全に返信する必要はないと考えている.

3.7 今後の課題

本節では, 本研究の今後の課題について述べる.

3.7.1 B-DRIP

本節では, 実装した B-DRIP の課題について述べる.

ネットワークアドレスでのルール設定

実装した B-DRIP では, ルールファイルでの閾値等の設定で, IP アドレスのみで指定するようになっている. B-DRIP を外部ネットワークと内部ネットワークの境界に設置する場合, 組織内にあるサーバやホストの IP アドレス毎にルールを設定するのは非常に手間である. 手間を省くために, 今後はネットワークアドレスを用いてルールを設定できるようにすることで, 管理者の負担を減らすことが求められる.

フラグ情報によるカウント

アクセスが攻撃かどうかの検出は, 単位時間あたりのパケット数のみを用いている. Flood 攻撃等の大量パケットによる DoS 攻撃への対処は, 現在のシステムで可能であるが, フラグ情報に基づいたカウントをすることで検出可能な攻撃を増やすことが可能である. 例えば, SYN フラグのみがたったパケットをカウントすることで, Brute Force Attack を検出することが可能になる.

閾値の設定方法

ルールファイルでは, 1 つの IP アドレスに対して 1 つの閾値を設定できる. しかし, Web サーバでは, アクセスするコンテンツによって送信されるパケットの数も大きく変化する. 閾値が 1 つの場合は false positive の可能性も大きくなってしまう. 今後は, URL 毎の閾値設定ができる機能を実装する必要がある. また, 現在は管理者が手動で閾値を設定する必要があるが, 自動化の機能を実装することが望ましい.

ネットワーク内の設置場所による評価

実運用を開始した B-DRIP は, 現在, 和歌山大学の DNS サーバの上位と，対外接続部に設置している. B-DRIP を外部ネットワークとの境界に設置することで, 学内ネットワークをすべて防御対象にすることができる. 境界に設置する場合は, ブリッジングするパケットが急増するため, B-DRIP の動作をさらに検証, 評価する必要がある. 併せて，対外接続におけるバーストトラヒックへの対処が可能かどうかを検証する必要がある.

3.7.2 TCP 代理応答サーバ

本節では, 実装した代理応答サーバの課題について述べる.

返信するプロトコルの増加

実装した代理応答サーバは, HTTP のみに返信可能である. 今後は SSH などの, 対話型プロトコルへの返信も可能にすることで, SSH の Brute Force Attack などを実行している攻撃者に対して, 警告コンテンツを返信することが可能となる.

返信性能の向上

実装した代理応答サーバは, 3000[pps] 以上の返信性能をもたないことが判明した. 代理応答サーバの目的は, 警告コンテンツを返信することであったため, 機能的な問題は無いが, 攻撃の規模によっては性能面も強化する必要があると考えられる. その場合，マルチスレッド化などの実装により, 返信性能を高めることが考えられる.

3.8 結言

本研究では，既存研究とは異なるソースアドレスルーティングによってさまざまな DoS 攻撃に対応できるシステムの提案と実装，および検証の結果を示した．先行研究である DRIP を改良し，透過型として既存ネットワークへ容易に導入が可能である．さらに，大量パケットの量的要因によって成立する DoS 攻撃に対して，過剰な検知である false-positive のフィードバックとして，代理応答サーバの併用を提案し，さまざまな性能評価を実施して有用性を示した.

今後の課題として以下の点を明らかにした．まず DoS 攻撃におけるパケット流入量を基に閾値を設定するのではなく，サーバからの返信パケットに着目した対応手法である．さらにサーバの負荷によって対応を開始する仕組みである．現在では，コンテンツの肥大化によってサーバ側の受信パケットより返信パケットの方がサーバに負荷をかけている．DoS 攻撃との判定が困難となるパケットの量的状況において，サーバ資源の保護を目的として，副次的に DoS 攻撃を排除するシステム設計と構築に臨む所存である.

第 4 章

情報セキュリティ技術者・管理者育成を目的とした情報危機管理演習の環境構築とその運用支援

4.1　序言

情報技術が世界的に普及整備され，経済社会の基盤となった現在，大量の情報がいつでもどこでも簡単にやり取りされるようになっている．これにより我々は保有する情報資源を有効にまた便利に活用できるようになっている．

一方でこれらの進んだ環境が悪用される例も後を絶たない．インターネット上でサービスを提供するサーバへの不正侵入，情報漏洩，個人情報の流出，ホームページ改竄，およびウイルス感染などが挙げられる．これら情報セキュリティ関連の事故・事件は問題を起こした当事者，被害にあった当事者だけのものにとどまらない．当事者らの周囲の取引相手や，経済取引そのものへの信用問題，社会インフラの機能麻痺など，社会経済全体に影響を及ぼす問題へと拡大していく．これまではネットワークや情報システムの事故に対して「未然に防ぐにはどうすればよいのか」が注目されてきた．

しかし，上記のような攻撃に対するリスクの拡大に対応するためには，「情報セキュリティに絶対はなく，事故は起こりうるもの」という前提にたった対策が必要となっている．きちんとした情報セキュリティ対策ができていれば，これらの被害を小さくすることができた可能性があるからである．

このことから現在では，情報の取り扱いにおける対策は「情報セキュリティ事故は起こりうるものとして，仮に起こった場合でも被害を低減，最小化し，早期に復旧させる」

ための仕組み，すなわち「事故前提社会システム」としての情報インフラ構築，および運用，対策に切り替わってきている [42]．そして，コンピュータネットワークの情報セキュリティを脅かす攻撃，事故が年々増大，複雑化，高度化する中，情報セキュリティ教育の実施や情報セキュリティ対策に必要な人材育成の必要性が高まっている．「事故前提社会システム」としての運用・対策は必要であるが，この実践は，事後の状況を個々の環境であらかじめ作り出すこととなり，現実的に困難であり，人的，物的にもコストがかかる．したがって事前に事後対策の実践に取り組むことができる情報セキュリティ演習の役割は非常に大きい．しかし，これまで行われてきた情報セキュリティ演習は，「事故前提社会システム」のための人材育成を考えると，顧客対応，広報活動等を含めた総合運用管理能力の育成への配慮に欠けている．また，一般的に参加者側の観点では，参加資格の有無，人数制限，演習会場への移動コストがかかり，主催運営側の観点では，会場での環境構築を行わなくてはならない．また，演習環境は複雑であり，機材の移動にもコストがかかる．そのためイベント的な活動になってしまい，定常的な運用が困難になっている．これらの理由から，演習実施・参加することそのものが難しいという問題点がある．

本論文では「事故前提社会システム」に対応できる情報セキュリティ技術者・管理者育成を目的とした情報危機管理演習の環境構築とその運用支援，とりわけ遠隔参加環境の構築手法を提案する．遠隔参加環境の構築では ASP(Application Service Provider) 化を目指し，遠隔から誰でも参加できる形式を実現することで，上記に挙げたコストが軽減され，情報セキュリティについての啓蒙を深めることができる．次節では，まず既存のセキュリティ演習と提案する「事故前提社会システム」を考慮に入れた演習環境，「情報危機管理演習」について述べる．その後，情報危機管理演習を含めた既存の演習全体での問題点について言及し，解決するための遠隔参加環境構築について提案を行う．また実環境で運用していく中で得られた成果・課題について考察し，将来の運用支援と拡張性について述べる．

4.2 既存の情報セキュリティ演習及びイベント

ここでは既存の情報セキュリティに関する演習で行われてきた類似する仕組みについて述べる．

情報セキュリティに関するイベントとして，まず IT Keys で行われている情報危機管理演習と似た演習である「インシデント体験演習 [46]」，ブラックハットが開催しているセキュリティ技術実習トレーニング「Black Hat Japan Training[47]」と日経バイトが主催した「セキュリティ・スタジアム [48]」の 3 つをとりあげる．

4.2.1 インシデント体験演習

インシデント体験演習は 4.3 節で述べる IT Keys の実践科目群の 1 つである．情報通信研究機構・北陸リサーチセンターの大規模汎用ネットワーク実証実験施設 StarBED を利用したセキュリティテストベッドをつくり，現実的な規模と複雑さをもつサイトへのさまざまな攻撃と，それらに対する監視・分析・防御・回避・復旧等の技術を実践的に体験習得するというものである．

実習システムが生成する，スキャン，DoS 攻撃，Worm 感染，自サイト内のボットの発見と対処，フィッシング被害の検出，P2P クライアントの検出等のインシデントに対して体験し，処理内容のレポートをその都度提出するという演習である．

4.2.2 Black Hat Japan Training

Black Hat が毎年 1 回日本で開催している Black Hat Japan Briefings & Training の中で執り行われているものである．このトレーニングでは，世界有数の情報セキュリティ専門家によって，内部者による個人情報等のデータ窃盗の証拠確保や捜査，マルウェアソフトに対抗する技術，シスコ製品への攻撃すなわちイントラネット攻撃に対する防衛技術，Web アプリケーションの脆弱性対策などについて学ぶコース等が用意されている．これらのコースには修了書，認定書が発行されており，受験者は相応の資格が取れるようになっている．例えば米国国家安全保障局（NSA）InfoSec アセスメント方法論（IAM）：レベル 1 などがある．また，企業や政府等多様な組織にあわせてカスタマイズされたプライベートトレーニングの手配も可能になっているため，応用の範囲は大きい．現在，2009 年度は中止されており，2010 年度以降も開催予定は不明となっている．

4.2.3 セキュリティ・スタジアム

セキュリティ・スタジアムは特設ブース内に外部ネットワークと分離された仮想のネットワークを構築し，そのネットワーク内に攻撃・防御・検知の 3 つのグループに分かれてお互いの技術を競うイベントである．

攻撃側は実際の不正アクセス技術を利用して攻撃を実施する．防御側は攻撃側の攻撃からコンピュータを守るために防御を行う．検知側はネットワーク上で不正アクセスがあったかどうかなどの監視を行う．それぞれが「相手の管理者権限を取得」，「侵入された」，「それらの行為をすべて記録した」等の結果を出した場合にそれを審判担当者に報告し，

検証を行う．

会場では実際に攻撃側が攻撃している画面等が表示されたり，時間帯によっては説明が行われるので，参加していない人でも興味をもつことができるようになっている．

4.3 情報危機管理演習

筆者らは「事故前提社会システム」に必要とされる情報セキュリティ技術者・管理者育成を行うための情報危機管理演習を提案し，実施してきた．以下にその内容を述べる．

情報危機管理演習は IT Keys 先導的 IT スペシャリスト育成プロジェクト [43] の実践科目の 1 つの「IT 危機管理演習 [44]」として行われている．IT Keys プロジェクトは情報セキュリティ分野における世界最高水準の人材育成拠点の形成を目的とする「文部科学省：平成 19 年度先導的 IT スペシャリスト育成推進プログラム」の 1 つとして，平成 19 年 10 月に 3 年半の予定でスタートしたプロジェクトである．もともとは「第 12 回サイバー犯罪に関する白浜シンポジウム」内で行われていた「危機管理コンテスト [45]」が名前を変えて採用されたものである．

過去 5 年間で筆者らのグループでは危機管理コンテストで予選を含めて 7 回，IT 危機管理演習 2 回の計 9 回において当演習環境構築，演習の運用管理を行った．情報危機管理演習は「事故前提社会システム」で必要とされる人材育成の考えから，実際に事故（インシデント）を起こされたものについて，単純なトラブルシュートの能力を問うにとどまらない．実際に IT サービスを提供している環境において，サービスを運用する側にとっては，法令遵守 (コンプライアンス) に基づいて，安易にサービスを停止，設定の変更および再稼働できないことがある．これらを実施するには，責任者の許可を得るなどの段取りが必要である．さらに，ユーザとのコミュニケーションによって状況を把握する一方で，運用側の内部事情の開示には細心の注意を払う必要がある．このように，本演習には総合マネージメント能力に育成に必要な要素が含まれている．以下にその流れについて述べる．

4.3.1 情報危機管理演習の流れ

情報危機管理演習では，実際に起こりうるインシデントとその事後処理について情報システム管理者の立場からロールプレイ形式で実習する．

演習は，トラブルシュート，顧客対応，広報，マネージメントを行う参加側が管理する各ブースと管理者，攻撃役，被害者役を行う運営側に分かれており，これに審査委員を加えて次のような流れで行われる．

1. 各ブース環境にはセキュリティホールが仕込んである
2. 運営側より何らかの方法で攻撃を発動して，仮想企業のサーバなどに障害を発生させる
3. 運営側が被害者役として障害を確認し苦情を送信する
4. 各ブースが障害に対応すると同時に，苦情への対応を行う
5. 障害を復旧させた後，一連の経過をトラブルチケット化して管理者に渡す
6. 障害対応の手際、トラブルチケットの報告をもとに審査委員が優秀チームを選定する

以下に，順を追って説明する．

演習環境にはあらかじめ運営側によりインシデント，あるいはこれを発生させるためのセキュリティホールが内包されている．しかし，この段階ではすべてのサービス，ネットワークは通常通りに稼働している．参加者は 5 人程度のグループとなり，仮想企業の情報セキュリティ担当を演じてもらい（図 4.1），運営側が起こすインシデントに対応してもらうため各ブースにて待機している．

機を見て運営側が参加側のブースに対して，事前に用意しておいたセキュリティホールに攻撃を発動し，サービス，ネットワークに障害を発生させる．これらのインシデントは一般では法律に接触するようなシナリオも多く，今後の復旧までの広報的な対応策等については情報セキュリティに関する法律や，セキュリティポリシーなどにも感心を寄せる必要がある．

さらに，運営側はネットワーク上のトラヒック管理を中心に参加チームの対応状況を把握するとともに，顧客あるいは外部ユーザを演じて適宜問い合わせや苦情のメールを送信する．なお，これら一連の行動記録は運営側で進行表に記録されている．

この段階にきて参加チーム側ははじめて障害の発生を知ることになるため，必然的に障害発生後，つまり事故前提の障害対応を求められることになる．

参加チームには発生したインシデントの障害対応に加えてトラブルチケットの提出までが求められている．

最後に，審査委員が，復旧までの時間経過，インシデントに対応した手法，および問い合わせや苦情に対する対応などを総合的に判断し，各参加側の障害対応を評価する（図 4.2）．

以上のように情報危機管理演習では，トラブルシュートの技術力はもちろん，情報分野の基礎学力の向上，総合的運用管理能力，それに伴うセキュリティポリシー，法律への関心を引き出すことができる．つまり，本演習は，情報セキュリティの総合マネージメントができる人材育成の場である．

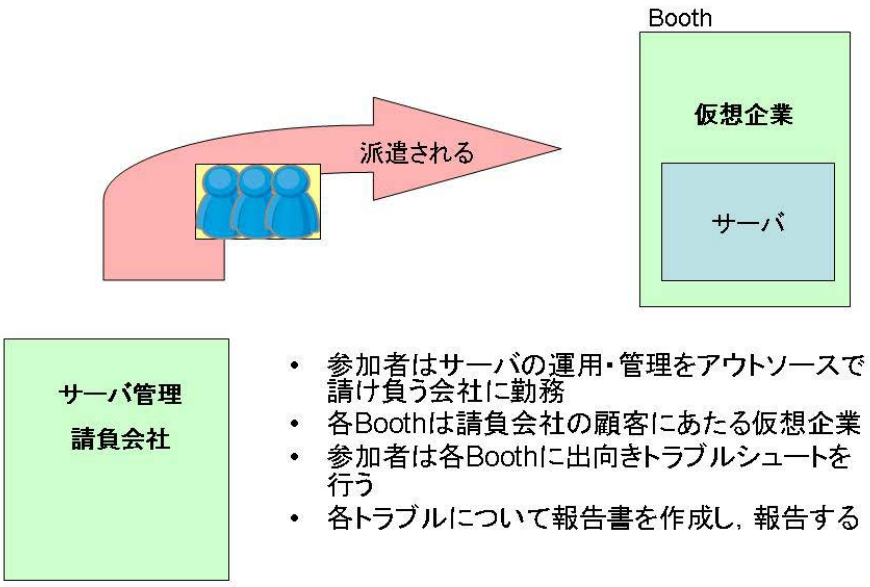


図 4.1　参加側シチュエーション

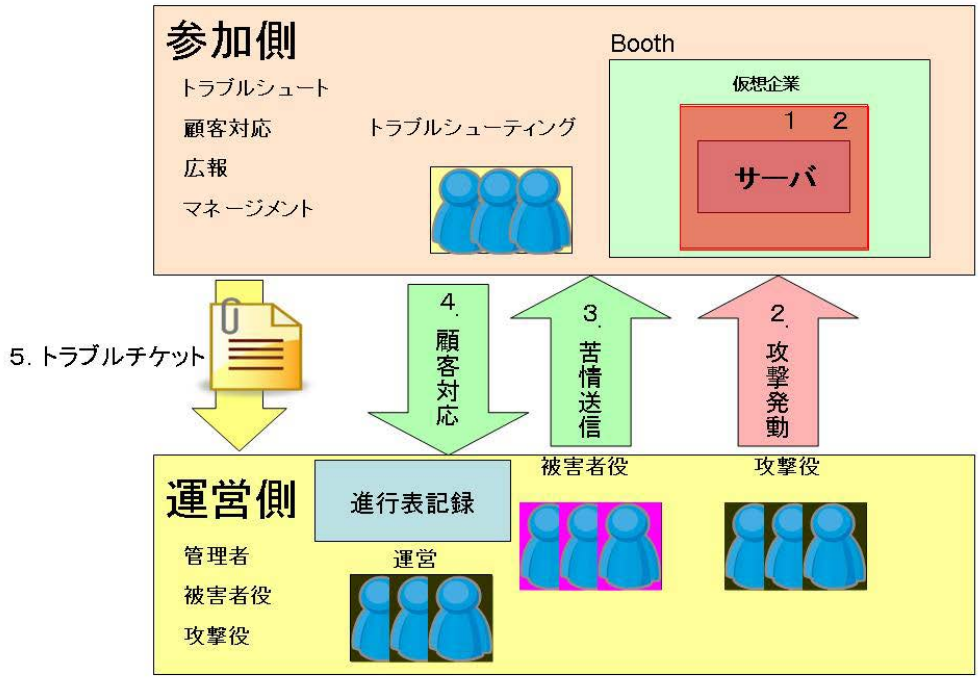
進行表，トラブルチケットについては以下で詳しく説明する．進行表とトラブルチケットの情報を合わせることによって，正確な演習の状況把握と評価ができる仕組みとなっている．

進行表

進行表は，運営側が取得した情報を時系列に沿って記録したものである．取得する情報として，主に参加側に対して行った行動，参加側が起こした行動などが挙げられる．これに加えて，審査委員が実際に参加側がトラブルシュートをしている現場を見ることにより得られた情報も随時加えている．図 4.3 に進行表の例を示す．

トラブルチケット

トラブルチケットは参加側がインシデントの発生から終了までの出来事を記録するものである．記録するものは，インシデントの発生した時刻，対応担当者，発見者を基本に，発生したインシデントの内容，復旧までに行ったトラブルシュートの手順，顧客対応の流れ，広報への対応方法の提案，原因の究明に至るすべての事項である．トラブルチケット



数字は 4.3.1 節の箇条書きの項に従う

図 4.2　情報危機管理演習全体の流れ

は演習での評価項目としてだけでなく，実際の運用の現場においても，トラブルチケットを残すことによって，インシデントについての情報共有を円滑にし，障害対応の見落としを少なくすることができる．また以降に同様のインシデントが起こった際に迅速な対応が可能になるといった利点がある．図 4.4 にトラブルチケットの例を示す．

4.4　既存環境の問題点

前節までに情報危機管理演習を含めて 4 つの情報セキュリティに関する演習を紹介した．

情報危機管理演習によって情報セキュリティの総合マネージメントができる人材育成の場を提供できるようになったが，依然としていずれの情報セキュリティに関する演習も参加するのが難しいという問題点が残る

IT Keys で開講されている「IT 危機管理演習」，「インシデント体験演習」に参加するためには IT Keys プログラムに参加する必要がある．IT Keys プログラムに参加するた

図 4.3　進行表例

めには参加条件があり，対象は，奈良先端科学技術大学院大学、大阪大学、京都大学、北陸先端科学技術大学院大学の各大学院修士課程（博士前期課程）の学生もしくは科目等履修生である．さらにプログラム参加の定員が 20 名程度となっており，これから新規に参加するのは困難であると考えられる．

「IT 危機管理演習」と同等の演習ができるサイバー犯罪に関する白浜シンポジウム」内で行われていた「危機管理コンテスト」に関しては，IT Keys にあった参加条件のようなものはない．しかし，ネットワーク機器の数の関係から，6 チーム（最大 30 人程度）の参加が限度となる．

「Black Hat Japan Training」は事前予約により参加が可能であるが，用意される座席数により，若干定員の制限が加えられる．またコースによって異なるが，参加資格の条件として，情報セキュリティ分野での実務経験を挙げるコースもある．さらに，この演習において参加が難しい理由はその参加費である．コースによって 20 万〜35 万円の参加費が発生し，費用面の制約から参加することは難しい．

「セキュリティ・スタジアム」に参加するには攻撃・防御・検知のいずれかの側に則ったサーバを用意する必要がある．しかし，この演習は 2004 年を最後に行われておらず，

--Booth2

1.障害の概要 ID :01

発生時刻 :10:36

発生報告 発見者 :イマイ　様

担当者 :カスタマー担当 中村

内容 :
ホームページへアクセスしたところ、異なるページへ転送されてしまった。転送先は、和歌山大学で
あった。

-- 2.障害への対応 [2009-09-05]
[10:34]イマイ様より上記内容の障害報告をメールで受理。
[10:36]イマイ様より上記内容の障害報告を電話で受理。
[10:38]障害の確認。
[10:39]イマイ様宛へ原因究明の旨のメールを送信。
[10:41]apache内の◆◆に、和歌山大学 (http://www.wakayama-u.ac.jp)が指定されているのを確認

[10:50]CEOへ障害の発生を連絡。
[10:51]対象箇所の削除を実行し、問題を解決した。
[10:52]イマイ様へ対応に関するご連絡の電話。
[10:53]イマイ様へ問題解決の報告メールを送信。

-- 3.原因の究明
設定ファイルを確認した結果、和歌山大学のページへ転送する設定になっていた為、
CEOの同意を得て該当箇所の削除を行い問題の解決をした。

図 4.4　トラブルチケット例

現在参加することはできない.

またこれらの演習，イベントは主催運営側が会場での環境構築を行わなくてはならない．演習環境は複雑であり，機材の移動にもコストがかかる．そのためイベント的な活動になってしまい，定常的な運用は不可能となっている.

このように，どの演習においても「演習に参加することが困難である」という問題点が依然残っている.

4.5　遠隔環境作成とその目的

前節では，これまでの情報セキュリティに関する演習に関して，いずれの演習においても参加することが困難であるという問題点があることがわかった．1 カ所の会場で実施されるため現地に赴く必要があること，多数の参加者を現地に参加させる上で，費用や設備が大規模になってしまうこと，および復旧作業を実施するために環境構築に配慮する必要があることなどが障壁となっている.

そこで筆者らは，情報危機管理演習を踏襲し，遠隔参加を前提とした情報危機管理演習

の環境構築及び運用支援を提案・実装し，情報セキュリティ対策のための ASP 化を目指した．

遠隔参加システムの利点として以下の点が挙げられる．参加者側としては参加資格の必要ない情報危機管理演習が遠隔参加できる形式で実現されることで移動コストが軽減され，場所に依存することなく演習を受けることができる．したがって，情報セキュリティ対策に必要な人材の育成にもつながるため「事故前提社会システム」の対応として成果が期待できる．運営側としてはサーバ，ネットワーク機器を演習会場に持ち込む必要がなくなり，こちらも演習会場に移動する必要がなくなるため，演習に携わるための移動，運搬費等のコストも軽減することができる．

筆者らが目的とする遠隔から演習環境に参加できる形で，大規模な情報セキュリティ演習が執り行われた例は，筆者らが調査した結果では見あたらない．しかし，現在の情報セキュリティの背景からなる情報危機管理演習の必要性から考えると，ASP 化された遠隔参加可能な情報危機管理演習は参加者のコストを抑えられるため，事故を前提とした情報セキュリティの対策に大変有用であると考えられる．

4.6 システム設計・技術要件

本節では，遠隔参加環境を前提とした情報危機管理演習の環境設定について述べる．システム設計の技術要件として，OS（Operating System）の仮想化，遠隔地からのアクセス技術として，管理側と遠隔参加側で安全にデータ転送するための暗号化，トンネリング技術，および双方の音声連絡用のシステムである IP 網を利用した音声通話技術の 3 つを挙げる．

4.6.1 VMwareServer

演習においてはインシデントの起こる状況は毎回同じでなくてはならない．なぜなら，演習としてインシデント対策を行う場合，複数回にわたって演習を行う際にも逐次人の手でセッティングをしていたのでは異なる環境が出てきてしまう恐れがある．状況が異なると評価基準がばらばらになり，評価にコストがかかる．よって毎回すべての環境が同じものにできるようにサーバ OS を仮想化し，どの端末でも全く同じ環境を作り出す必要がある．

仮想化には VMware 社の VMware Server[49] を使用する．さらにこの技術によって，演習後に仮想化された OS データを保存することができるので，いつでもその状況を再現

し見直すことができる．

また提案する遠隔参加環境においては利用者端末で VMware Server Console というアプリケーションを利用する．これにより，現場にいなくとも仮想サーバの電源管理が可能になる．

4.6.2 PPTP

PPTP (Point-to-Point Tunneling Protocol)[50] は，TCP/IP ベースのデータネットワーク上に仮想プライベートネットワークを作り出すことにより，リモートユーザから企業のサーバへの安全なデータ転送を可能にするためのネットワークプロトコルである．

PPTP は他の VPN (Virtual Private Network) 技術に比べてクライアント側で Windows や MacOSX といった多くのシェアを占める OS にソフトウェアが標準搭載されているため，新たなソフトウェアの導入が不要であるという利点がある．

運営側と遠隔参加側とをデータ転送するためのトンネリング技術には，PPTP サーバに MPPE（Microsoft Point-to-Point Encryption）暗号化方式，MS-CHAP-V2（MicroSoft Challenge Handshake Authentication Protocol version 2）認証方式を実装する．これにより運営側（サーバ側）と参加チーム（クライアント側）との間を安全に通信することができる．

4.6.3 Skype

情報危機管理演習の場合，運営側，遠隔参加側双方の音声連絡用のシステムが必要になる．

従来のネットワークが演習会場内で収まる形での情報危機管理演習では音声連絡に Cisco 社の CallManeger を使用していたが，参加側がインターネット越しに遠隔参加する環境では，グローバルネットワーク下でも使用勝手のよい Skype Technologies 社が提供するインターネット電話サービス Skype[8] を使用する．

Skype は P2P 技術を応用した音声通話ソフトである．Skype をインストールしてユーザ登録し，パソコンにマイク (と必要に応じてヘッドフォン) を接続すれば，ユーザ同士で音声による通信を行うことができる．また同時通話，テキストによるチャットやファイル転送などもできる．インスタントメッセンジャーのように通話相手を「友達リスト」で管理することができ，オンライン状況をリアルタイムに確認することができる．また，Skype は IP 電話などと異なり，中央サーバを介さずユーザ同士が直接接続して通話する．

ファイアウォールや NAT(Network Address Translation) の内側にあるパソコンからも，特別な設定を行うことなく接続できる．また通信内容は 128 ビットの AES(Advanced Encryption Standard) で暗号化されるため安全性も確保できる．

4.7 システムの実装

本節では，遠隔参加者が演習環境に接続するためのゲートウェイサーバを構築する．またそのゲートウェイサーバを組み込んだ情報危機管理演習環境ネットワークの実装，及び作成したシステムの動作について述べる．

4.7.1 既存ネットワーク構成

まず，情報危機管理演習環境でのネットワーク構成を図 4.5 に示す．この構成で使われているネットワーク機器は以下のようになっている．

- YAMAHA:RTX1100
- Cisco:Catalyst 3560
- Cisco:Catalyst 2940

これらの機器の役割としては，上位の RTX1100 がグローバルセグメントからのグローバルアドレスを NAT 変換する．Catalyst 3560 が運用側セグメント，Booth セグメント（参加者側が使う）を含めたルーティングを行う．また，下位の RTX1100 が Booth セグメントのルーティングを行う設定になっている．なお，Catalyst 2940 は L2 スイッチとして使用している．

また図 4.6 の Booth1 の例に示すように，Booth セグメント内において RTX1100・Catalyst 2940 以下にあるセグメントのうち，一方のセグメントが用意されたサーバが存在するセグメント（サーバセグメント）である．用意されるサーバは VMwareServer を使用し仮想化している．具体的には WindowsServer2003 上で VMwareServer が起動しており，その中で 2 つの情報危機管理演習用サーバが動いている．もう一方のセグメントが，参加者側がそのサーバを管理するためのセグメント（参加者側セグメント）になっている．このセグメントに参加者が PC 端末を接続することで，それぞれのネットワーク (10.1.0.0/16) に通信できるようになり，危機管理演習に参加できる．

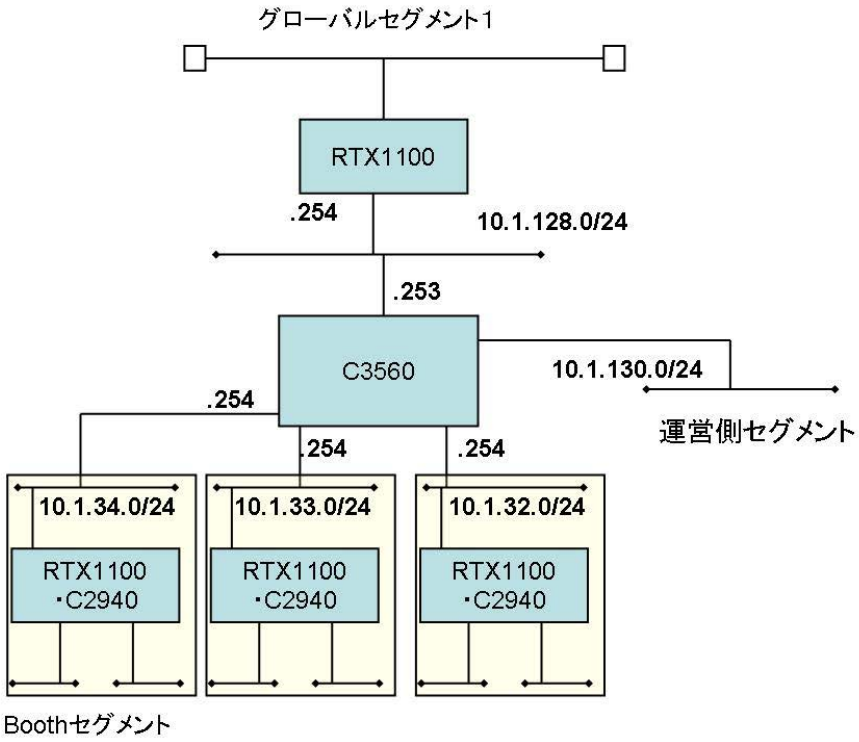


図 4.5　情報危機管理演習環境のネットワーク

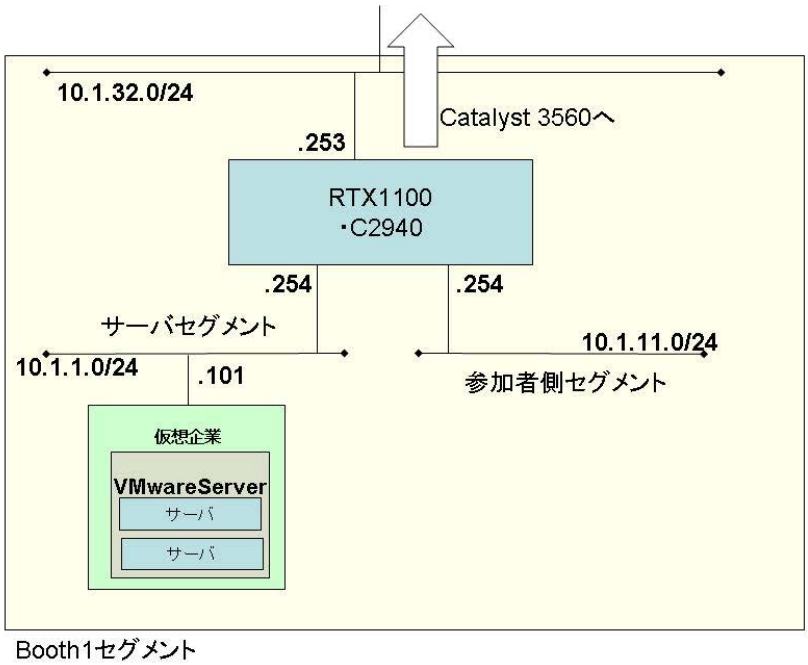
4.7.2　提案ネットワーク環境

前項のネットワークにゲートウェイサーバを組み合わせ，遠隔参加可能な情報危機管理演習環境を構築する．

ネットワーク概要としては図 4.7 に示すように，運営側に既存の情報危機管理演習環境を構築し．この環境に新しく PPTP ゲートウェイサーバ（GWserver）を設置する．参加者はこのゲートウェイサーバに対してインターネット越しに接続，認証を受けることによって演習環境に接続することができる．

具体的には遠隔からの参加者が，PC 端末を接続して危機管理演習に参加するための参加者側セグメント (10.1.11.0/24) にゲートウェイサーバのインタフェースの一方を接続する．

もう一方のインタフェースは情報危機管理演習環境ネットワークが存在する「グローバルセグメント 1」と別の「グローバルセグメント 2」に接続するように構成した．「グローバルセグメント 1」は参加者側がインターネットから PPTP 接続するためのネットワー



Booth1 セグメント周辺部

図 4.6　Booth1 セグメントのネットワーク

クであり，「グローバルセグメント 2」は Booth1 の各サーバなどがインターネットに出るためのネットワークである．Booth1 で発生させる障害には，Booth1 のサーバやネットワーク機器がインターネットに接続できない内容も含まれている．しかし，遠隔からの参加者が Booth1 に接続できなければ障害に対応できない．したがって，グローバルセグメントが 2 つある理由は，Booth1 からインターネットへの接続にトラブルが発生した場合でも，参加者側による障害対応を可能にするためである．

Booth1 を中心に一部抜粋した提案ネットワークは図 4.8 のようになる．

4.7.3　ゲートウェイサーバ

本システムで用いたゲートウェイサーバのソフトウェアは以下のようになっている．

- ハイパーバイザ:VMware ESXi3.5.0
- OS:FreeBSD6.3
- PPTP サーバ:PoPToP1.3.4

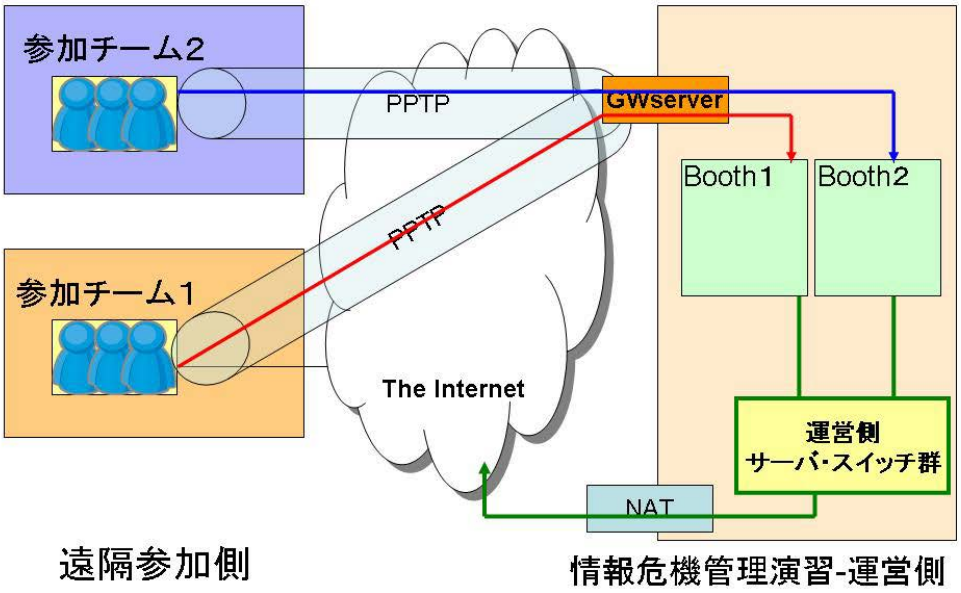


図 4.7　遠隔参加環境ネットワーク概要

- ファイアーウォール:ipfw

VMware ESXi は VMkernel と呼ばれる専用のホストカーネルによって直接ハードウェア上で仮想マシンの動作を制御している．VMkernel は最低限の機能のみを提供しており，画面などをもたない．そのため，ユーザが仮想マシンの制御などを行うことができない．そこで，ユーザに仮想マシンをわかりやすい形で制御できるように，Windows，Linux を用いた制御環境を提供している．今回は VMware ESXi を用いて，OS を仮想化することで，運用側の構築コストを低減するとともに将来への拡張に関しても幅をもたせた．

PPTP サーバソフトウェアには PoPToP1.3.4 を使用した．PoPToP は Unix 環境下で構築できる PPTP サーバである．MS-CHAP-V2，MPPE の認証暗号化などが利用可能であり，複数クライアントを同時に接続可能である．本環境では，認証には MS-CHAP-V2 方式，暗号化には MPPE128 ビット暗号化のみを使用し，遠隔演習環境を構築した．

本環境のゲートウェイサーバは，図 4.9 にあるように PPTP でトンネル接続を行う際の認証によって接続するセグメントを振り分ける仕組みを実装した．例えば，user1 がこのゲートウェイサーバに接続してトンネル接続を行おうとすると，ゲートウェイサーバの

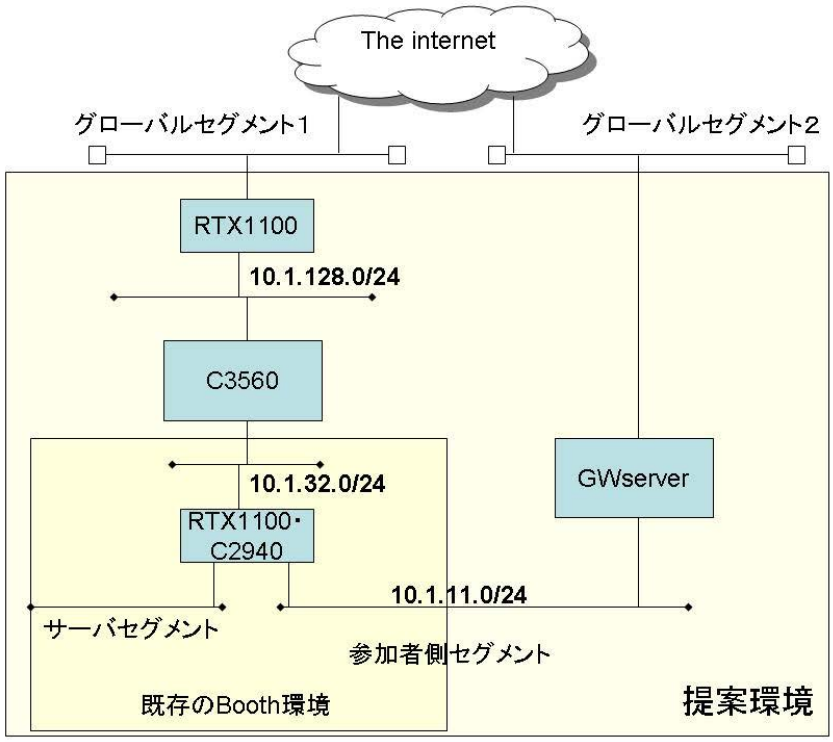


図 4.8　提案ネットワーク

認証テーブルにより，Booth1 で演習で行うための接続セグメントと，IP アドレスが割り振られるようになっている．同様に user2 が接続しようとすると Booth2 へのセグメントへ自動的に割り振られ，user3 ならば Booth3 に割り振られる．

また遠隔で PPTP 接続している仮想端末のルーティングは，ゲートウェイサーバ上のルーティングテーブルに影響される．そのため，単純なゲートウェイサーバのルーティングのみを利用しようとすると，例えば図 4.10 にあるように Booth1 に繋がった端末が運営サーバ・スイッチ群と通信を行う際に，本来 1 のルートを通る必要があるのに，実際には 2 のルートを通り，Booth2 のネットワークを通ってしまうという問題が起こる．

1. OS の標準ルーティングルール
   (a) 送信先 IP アドレスが Booth1 に属するならば Booth1 のサーバに送る
   (b) 送信先 IP アドレスが Booth2 に属するならば Booth2 のサーバに送る
   (c) 送信先 IP アドレスが運営側に属するならば Booth2 のサーバに送る
   (d) デフォルトゲートウェイは外向きへ送る

認証テーブル

ユーザ名	パスワード	接続セグメント	接続Booth
user1	12345679	10.1.11.0/24	Booth1
user2	34567891	10.1.12.0/24	Booth2
user3	amfiwjeni	10.1.13.0/24	Booth3

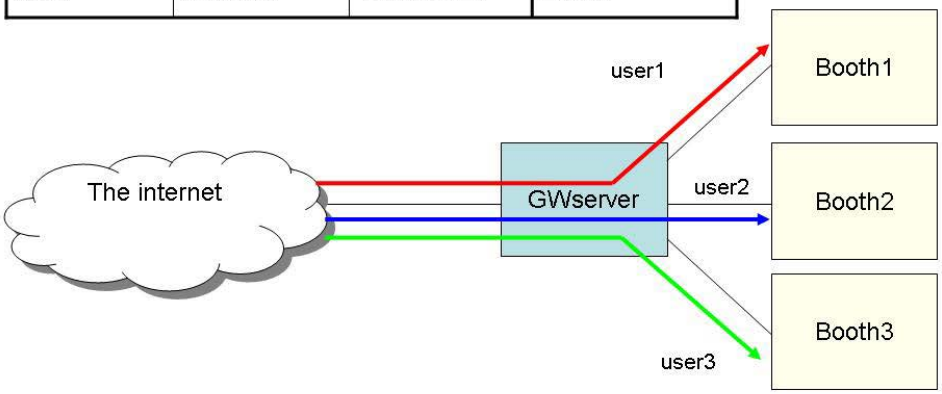


図 4.9　ゲートウェイサーバ

ゲートウェイサーバのルーティングテーブルの中で問題がある設定は，３番目の“送信先 IP アドレスが運営側に属するならば Booth2 のサーバに送る”という設定である．この設定のために Booth1 に繋がった端末では Booth1 のネットワークに障害がなくとも運営サーバ行きのパケットが Booth2 ルータへ送られてしまう．

ここで本システムでは ipfw を利用した．ipfw は FreeBSD に付随しているファイアーウォールであるが，動作機能の中にフォワードという機能が存在する．この機能を用いて送信先 IP アドレスだけでなく送信元アドレスを考慮に入れたルーティングルールを加えることで問題を解決した．

実際には“運営側行きのパケットは Booth2 スイッチへ送る”という OS 標準のルーティングルールを削除し，ipfw を用いて新しい条件を加えて新しいルーティングルールを作成した．

最終的なルーティングルールは以下のようになる．

1. ipfw によるフォワードによるルーティングルール
   (a) 送信元 IP アドレスが Booth1 に属するもので，送信先 IP アドレスが運営側

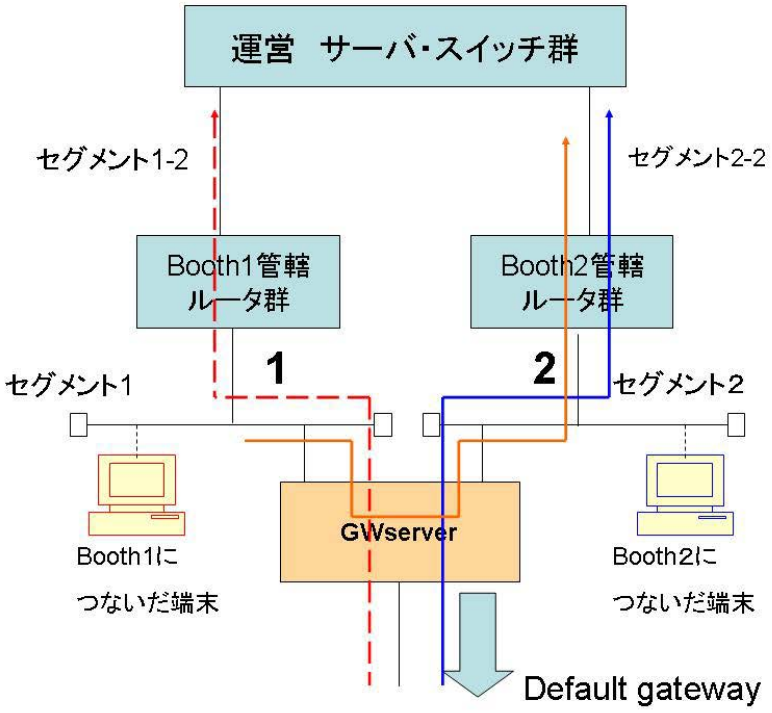


図 4.10　ルーティング問題

に属するならば Booth1 のサーバに送る

(b) 送信元 IP アドレスが Booth2 に属するもので，送信先 IP アドレスが運営側に属するならば Booth2 のサーバに送る

2. OS の標準ルーティングルール
   (a) Booth1 行きのパケットは Booth1 ルータへ送る
   (b) Booth2 行きのパケットは Booth2 ルータへ送る
   (c) デフォルトゲートウェイは外向きへ送る

上記のルールは，ipfw によるルールが優先，また上位にあるルールから適用される．これにより，ルーティング問題の解決を図ることができた．

4.7.4　システムの動作

実装されたシステムでの動作を図 4.11 にまとめた．まず参加側はグローバルセグメント 2 に繋がるゲートウェイサーバを通して PPTP トンネリング接続を行う (番号 1)．するとゲートウェイサーバによって自動的に IP アドレスが払い出され仮想的に参加者

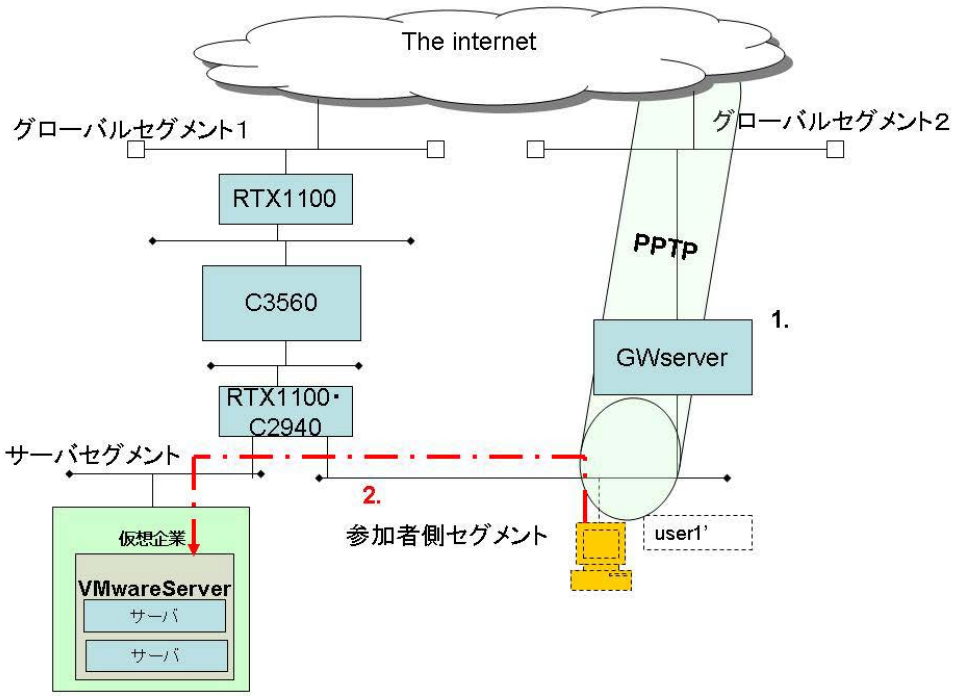


図 4.11　システムの動作

側サーバ管理用セグメントに接続することができる．トンネリング接続の後に VMware Server Console を用いてサーバセグメントの Windows サーバに接続する（番号２）．これにより，VMware Server 上の仮想サーバにアクセスすることができ，演習に取り組むことが可能となる．

図 4.12，図 4.13 に PPTP 接続時の詳細と VMware Srever Console で接続した際の表示の様子をそれぞれ示す．また，Skype についても PPTP 接続環境下で接続可能であり，音声通話も可能であることが確認できた．

4.7.5　提案環境の運用実績

実運用に関しては 2009 年 5 月に行われた第 13 回サイバー犯罪に関する白浜シンポジウムで行われた第 4 回危機管理コンテスト予選 [51] にて初運用を行い，8 チームが参加，2010 年 5 月に第 5 回同コンテスト予選 [52] にて 2 回目の運用を行い 6 チームが参加した．

予選参加者への接続方法の詳細説明などは参加者代表へのメールと Web を用いて行

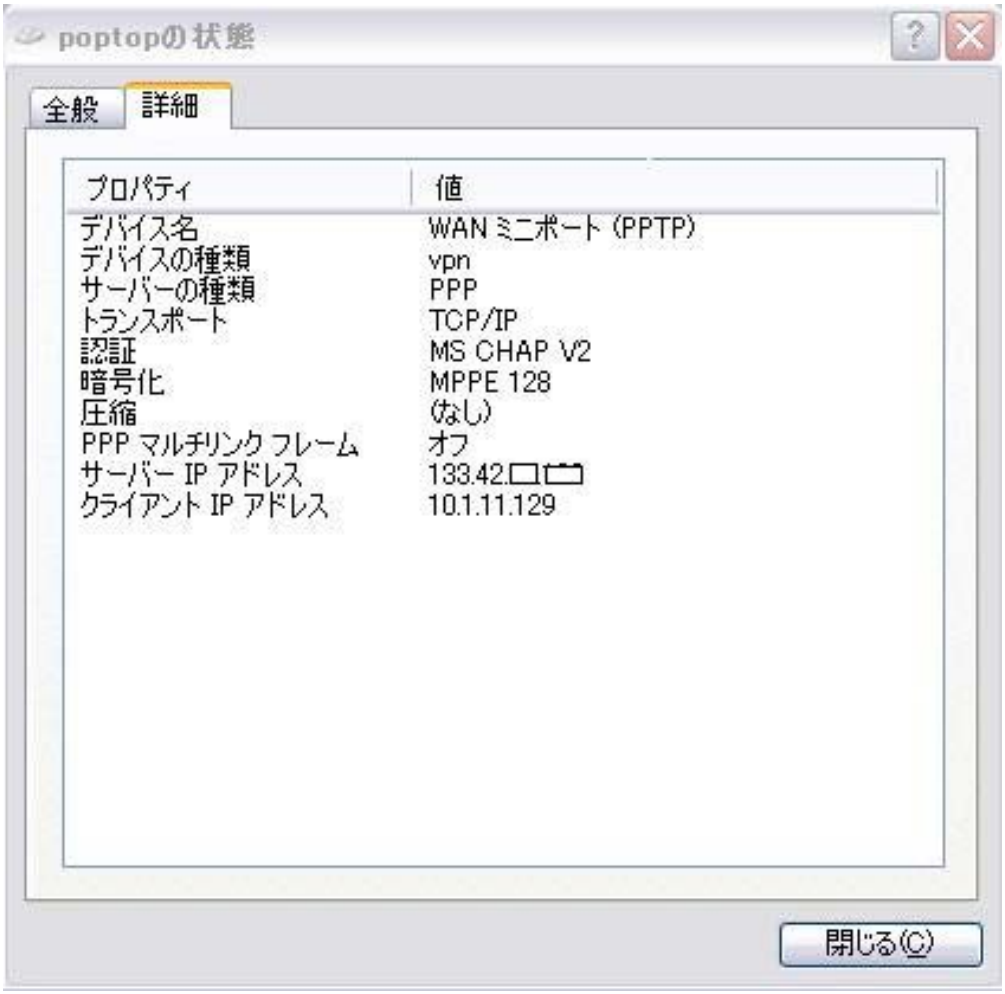


図 4.12　PPTP 接続の詳細

なった（図 4.14）.

演習を開始するにあたって参加側に行う初期設定が正しく設定されておらず，進行が遅れてしまう程度のトラブルは発生したものの，Skype を用いた音声連絡によって適切に対処できたことから，遠隔参加環境としての役割は十分に果たすことができた.

4.8　既存の演習及びイベントとの比較・評価

本論文で構築した遠隔参加を前提とした情報危機管理演習と，4.2 節で述べた既存の情報セキュリティ関連演習の違いについて考察し，本演習システムとの比較評価について述べる．以下に評価項目を挙げ，それぞれについて評価する.

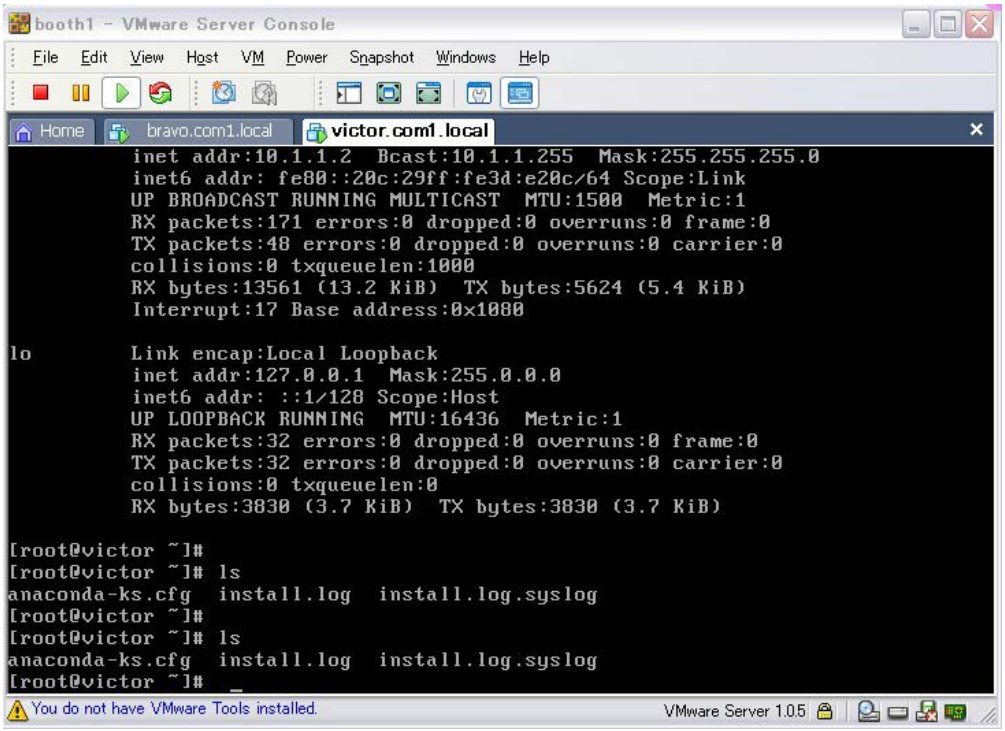


図 4.13　VMware Server Console

4.8.1　参加コスト

本項目では，参加者側，運営側の観点で，目的に挙げた参加コストに対する比較を行う．

参加者側の観点で比較すると，本研究により情報危機管理演習が遠隔で参加できるようになった．また，運営側の観点でも，会場までネットワーク機器等を運搬するコスト必要がなくなることから開催頻度を上げることができる．

したがって，本演習のシステムでは，既存の情報セキュリティに関する演習及びイベントに共通して存在した参加コストの問題が解消した．

4.8.2　ASP 化

本演習システムによって情報危機管理演習の ASP 化ができる．以下に ASP 化することにおいて重要となるポイントを挙げて比較する．

既存の情報危機管理演習と同じように，IP 電話，メールを用いた顧客対応を再現したことにより，参加者側は総合的な運用管理能力を演習下で養うことができる．また

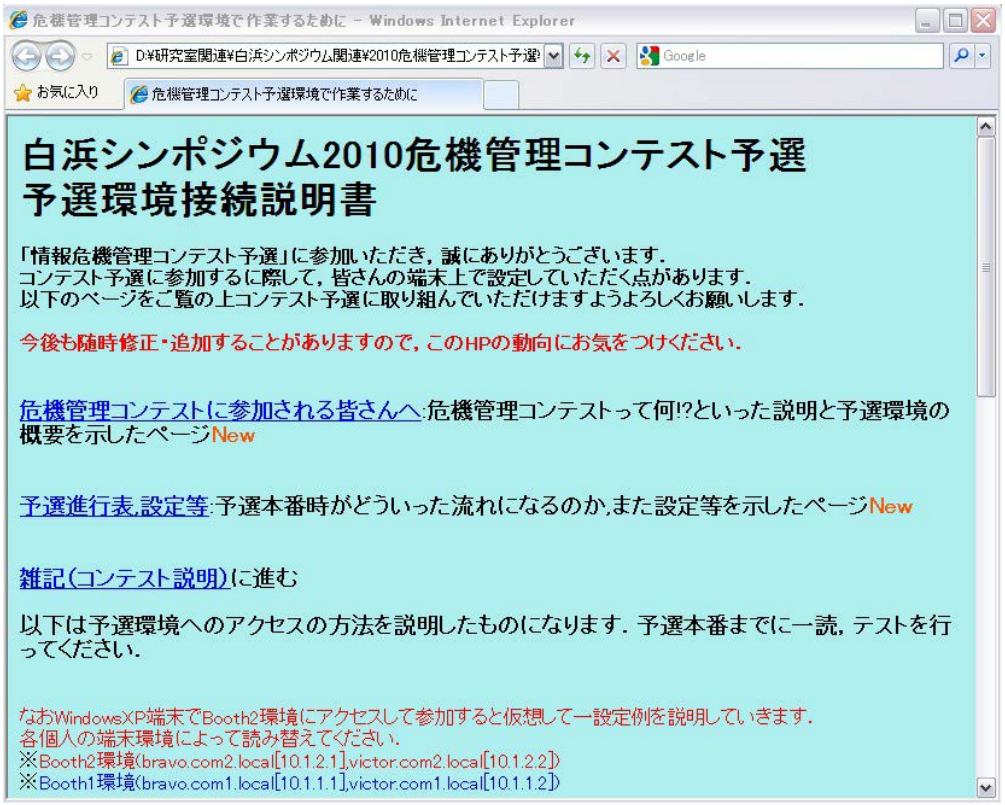


図 4.14　予選連絡用 HP

VMwareServer によるサーバの仮想化により，演習後に OS のデータを保存できるため，継続参加した際に，過去の結果と比べて自己の成長を推し量ることができる．

加えて，演習環境の初期状態を容易に復元できるため，審査委員は演習後の評価が容易になる．

また，運営側の観点ではローカル環境へのトンネリング接続による障害対応を実現しローカル環境で障害対応を行うことによって，参加者側から「外部からネットワーク機器を操作した結果，演習環境に繋げなくなった」，といったようなトラブルを防ぐことができる．

さらに，遠隔演習であっても複数参加者の対応を並行して，演習を進行させることを可能にしたことで，相互にスパムメールの踏み台攻撃を受けるなど，参加者同士のやり取りが必要になるようなシナリオを，演習プログラムに組み込める．

上記の点を遠隔参加環境においても可能としたものが本演習システムであり，構築した情報危機管理演習環境の場を ASP として広く外部に提供できるようになった．

4.8.3　トラブルシュートの技術力向上

本項目では各情報セキュリティ演習及びイベントでトラブルシュートの技術力向上の機会を得られるかという点を参加者の観点で比較する．

情報セキュリティ分野の演習なので，参加者にとってどの演習でも効果がある．Black Hat Japan Training はコースによってはトラブルシュートとは関係がないことがあるが，高レベルの情報セキュリティ技術を学ぶことができる．また認定書，修了書が出ることから確かな技術の証明をすることができる．しかし，セキュリティ・スタジアムに関しては，攻撃，防御，監視にチームが分かれること，さらにトラブルシュートが目的というわけではないので若干トラブルシュートの技術向上に結びつきにくい．

4.8.4　総合的運用管理能力の向上

本項目では各情報セキュリティ演習及びイベントで総合的運用管理能力の向上の機会を提供できるかという点を参加者側の観点で比較する．

情報危機管理演習では実際に「仮想企業，仮想被害者からの苦情への対応をする」というフェーズが存在する．これはインシデント体験演習，Black Hat Japan Training，セキュリティ・スタジアムのいずれの演習においても行われていない部分であり，これまでの「未然に防ぐ場合」には必要なかったが，「事故前提社会システム」を提唱する場合には，重要な部分になる．なぜなら，「未然に防ぐ場合」には，被害者は存在せず，情報セキュリティ担当者の能力としては，コンピュータに対する情報セキュリティ対策だけでよかった．しかし，「事故前提の場合」にはすでに事故が起こっている．そこには被害者も存在することになる．したがって，事故処理のための情報セキュリティ対策のみならず，総合的運用管理能力としての被害者への対応（顧客対応，広報活動）が必要になる．この点で情報危機管理演習は今求められている情報セキュリティ演習であるといえる．

4.8.5　法律に対するアプローチ

情報セキュリティ対策のためには，知るべき法律[*1]や選定するべきセキュリティポリシーがある．直接演習で学ぶ機会は少ないが，その類に関心が持てるような演習であるかどうかを参加者側の観点で比較してみた．情報危機管理演習では，上記の総合的運用

[*1] 不正アクセス行為の禁止等に関する法律など

管理能力の部分から，必然的に学ぶ機会をより多く提供できる．また Black Hat Japan Training でも，コースによっては学ぶことがある．インシデント体験演習, セキュリティ・スタジアムで，この点にふれることは比較的少ないと考えられる．

4.8.6 比較評価まとめ

前節までがこれまでのセキュリティに関する演習で行われてきた類似する仕組みと情報危機管理演習の比較になる．まず，表 4.1 に本研究を取り入れた情報危機管理演習と，とりあげた 3 演習との比較をまとめ，また本システムの欠点について述べる．

表 4.1 関連するセキュリティ演習との比較

比較項目	1	2	3	4
参加コスト	○	×	×	×
ASP 化	○	×	×	×
トラブルシュートの技術力向上	○	○	○	△
総合的運用管理能力の向上	○	×	×	×
法律に対するアプローチ	○	×	△	×

1…遠隔参加を前提とした情報危機管理演習（本演習システム）
2…インシデント体験演習
3…Black Hat Japan Training
4…セキュリティ・スタジアム

ここで表 4.1 のみを参照すると，本システムに欠点がないようにみえる．しかし，遠隔化によって欠点が発生することも判明している．

遠隔化の欠点としては主に「攻撃シナリオの種類が限定される」,「運営側・審査委員から参加側の様子がみえない」という 2 点が挙げられる．

前者については，遠隔で演習を行いネットワークに障害が発生したと仮定したとき，演習環境と関係ない外部ネットワーク（インターネット網）と，内部ネットワーク（演習環境ネットワーク）での問題の切り分けが難しく，混乱を来す可能性が高いと考えられる．

つまり，WEP クラック，WEB 認証，ネットワークのループなどといったネットワーク機器の再設定を要する攻撃シナリオ，さらにクライアント側に原因が依存する攻撃シナ

リオ（ブース環境下で起こるトラブル）については適用が困難であるといえる．

よって本研究の遠隔参加環境では，cgi,php の脆弱性を用いた WEB 改ざん，パスワード攻撃及び管理者権限奪取などのサーバ系インシデントを想定した攻撃シナリオに限定すべきであるといえる．

後者については，これまで同一会場内で行われていた演習を遠隔参加環境で実現したことによって演習拠点が複数になってしまった．このため，運営側はもとより，審査委員も実際に参加側のトラブル対応の様子を窺うことが難しくなり，運営側の演習進行，評価側の評価に影響が出てしまう．

4.9 考察・今後の課題

今回の研究で，比較的参加しやすい情報セキュリティの総合運用管理能力をもった人材育成の場を提供するシステムを構築した．このシステムにさらなる改良を加えることで，情報危機管理学習の場を貸し出すという ASP サービスの展開ができると考えている．

また，これまで情報分野での人材育成がそれぞれの企業や教育機関で行われることはあっても，それが公の場で行われることは少なかった．本遠隔参加環境によって情報危機管理演習の場を公にすることができるようになり，情報分野の人材育成にも貢献できているのではないかと考えられる．

しかし，このシステムを運用管理する上での課題もいくつか挙げられる，前節の比較評価で述べた欠点についてである．これに解決案を加えて以下にまとめる．

4.9.1 遠隔化の欠点

遠隔化の欠点としては前節で述べたように主に「攻撃シナリオの種類が限定される」，「運営側・審査委員から参加側の様子がみえない」という 2 点が挙げられる．それぞれ，前者はサーバ系インシデントを想定した攻撃シナリオに限定される，後者は参加者の動向が見えにくくなり，運営側は演習進行を円滑に行うのが難しくなり，評価側には評価が難しくなるといったものである．

4.9.2 遠隔参加環境の拡張

前節までに述べた問題点に対して現在始動している解決案について簡単に述べる．現在始動している対策として，演習環境に「複数のネットワーク構成の準備」をすることと演習中の参加者の残すログ，進行表，トラブルチケットなどを機能的にまとめ，表示するこ

とのできる「統合インタフェース作成」の２つがある．

前者の１つは 4.5 節にあげたインターネット網を利用した遠隔参加環境ではなく，第三者機関のイントラネットといった専用線を利用する方法である．演習機器のうち，サーバのみを運営側に残し，ネットワーク機器などは参加者側に設置する，といった構成をとることで，前項で実施困難とされたシナリオを使用可能にできると考えている．

後者については遠隔参加環境になったことで参加者への評価が困難となった．運営側，審査委員が既存の演習環境で行ってきたものと近い情報を迅速に得るため，参加側がそれぞれのサーバで行ったコマンドのログ，運営側とのメールのやり取り，進行表，トラブルチケットの情報を統合し，機能的にまとめたインタフェースを作成する．これは演習の記録という観点においても重要になると考える．

4.10　結言

本論文では，既存の情報セキュリティに関する演習及びイベントに対して，「事故前提社会システム」に対応できる情報セキュリティ技術者・管理者育成を目的とした情報危機管理演習の環境構築とその運用支援，とりわけ遠隔参加環境の構築手法について述べた．

演習への参加が困難であるという問題点を解決するために，情報危機管理演習の環境に PPTP ゲートウェイサーバを設置することによって，遠隔環境からの演習参加を可能にするシステムを構築した．また他の類似する情報セキュリティに関する演習と比較することで，遠隔参加可能とした情報危機管理演習が有用であることを述べた．

今後の課題として，遠隔参加環境における参加者側の状況を詳細に把握するシステムや，演習の評価指標として，障害の発生と対応状況によって仮想企業の株価を変動させるシステムの導入を検討している．これらは，参加者の評価を客観的に実施する上で有用であると考える．

第 5 章

結論

本論文では，ネットワーク運用管理技術と，運用管理の人材育成に必要な支援体制について，各研究の成果を明らかにしてきた．ネットワーク運用管理には，ネットワーク階層構造全体に対する広範な知識と技術が必要である．インターネットが世界中で急激に広まり，同時にさまざまなネットワーク技術とアプリケーションの開発が進んだことは，「他の階層の仕様や実装を気にする必要がない」という階層構造の利点が最大限に活かされた結果といえる．しかし，本論文で紹介したファイル共有ソフトウェアの利用検知や DoS 攻撃への対応をはじめ，各階層における挙動を正確に把握し，あるいは十分な対応能力を備えるには，1 つの階層における状況を把握するだけでなく，他の階層の情報を組み合わせることが重要である．さらに，運用管理の人材育成にも，既存の階層モデルに沿った技術だけでなく，コンプライアンスなどの法律的な観点やコミュニケーション能力を駆使する演習のフィールドが必要である．本論文では，上記に基づいた運用管理技術と支援体制について，実装と実運用を踏まえて提案した．

第 2 章では，ファイル共有ソフトウェア利用端末のトラヒックパターンを解析することで，ファイル共有ソフトウェアを検出する手法を提案した．本手法を実装したシステムは，現在和歌山大学の対外接続部で稼働中であり，セキュリティポリシで利用を禁止されている各種ファイル共有ソフトウェアの利用検知を継続して実施している．その成果を以下にまとめる．

- 従来の通信相手先数と非 well-known ポートに着目する手法に加えて，送信元ポート番号の連続性や同一性に着目し，ホワイトリスト・ブラックリスト判定と，UDP および TCP 判定により高精度な利用検知を実現した．
- パケットのペイロードを検閲することなく，トラヒックパターンのみを用いている

ため，プライバシへの最大限の配慮を施している．

- 有用な P2P ソフトウェアである Skype については，ホワイトリストで誤検出を防ぐ手法を提案し有効性を確認した．さらに，同一端末で複数アプリケーションが稼働する状況下の評価実験を実際の運用ネットワーク上で実施し，一般社団法人コンテンツ海外流通促進機構が平成 23 年 1 月に公開した調査に基づく全 15 種類すべてのファイル共有ソフトウェアの検出を確認するとともに，本システムの有用性を示した．これらは多くの既存研究で課題とされてきた事案である．

第 3 章では，SYN Flood 攻撃や UDP Flood 攻撃，および F5 リロード攻撃に対して送信元 IP アドレスを用いた防御手法と，特に F5 リロード攻撃への代理応答の仕組みを提案した．階層モデルにおけるネットワーク層の経路制御と，データリンク層の MAC アドレス変換を組み合わせることで，高い防御能力を備えるシステムであることを示した．本システムは，現在和歌山大学の対外 DNS サーバの上位と，対外接続部で稼働中である．以下にその成果を述べる．

- ソースアドレスルーティングによって攻撃者のトラヒックのみを逸らし，NULL デバイスに転送することで，SYN Flood 攻撃や UDP Flood 攻撃など大量のパケットを発生させる DoS 攻撃への高い耐性を示した．
- 仮想ネットワークインタフェースを用いるとともに，カーネルのソースコードを改良することでブリッジモードを実現した．これにより，サーバ群の上位接続部や対外接続部へ，ネットワーク構成を変更することなく容易に本システムを導入可能となった．
- 上記に加え，階層モデルのネットワーク層に実装したソースアドレスルーティングで，データリンク層の MAC アドレス変換を可能とすることで，F5 リロード攻撃などを代理応答サーバに転送し，警告コンテンツの表示による本システムのフィードバック，すなわち，false-positive 対応が可能であることを示した．

第 4 章では，既存の情報セキュリティに関する演習及びイベントに対して，「事故前提社会システム」に対応できる情報セキュリティ技術者・管理者育成を目的とした情報危機管理演習の環境構築とその運用支援，とりわけ遠隔参加環境の構築手法について述べた．以下に，その成果を述べる．

- IT スキルを競う CTF 形式と違い，IT リスクを低減するための要素として，IT スキルだけでなくコンプライアンスやユーザ対応など，法律やコミュニケーション能力を含めた総合的な人材育成の演習環境と運用支援を実現した．

- サーバ群を仮想化し，PPTP によるトンネル技術とソースアドレスルーティングによって，複数の参加チームが遠隔から当該演習に取り組める環境を実現した．
- シナリオに基づいた演習形式によって，参加側と運用側のインタラクションにおいて，運用側に明確な目的をもたせている，これにより，参加側の間違った対応によって発生する別の不具合を指摘して修正させるとともに，緊急対応の内容によってリスクヘッジの程度が変わることを体感させることが可能である．

以上のように，本研究で提案した手法は，ネットワーク運用管理技術の幅を広げるとともに，総合的な運用管理の人材育成に有効であることを示した．同手法が実際の運用や演習に導入され，継続的に実施されていることからも，その有効が裏付けられている．最後に，本研究に関する今後の課題としては，以下が考えられる．

- ファイル共有ソフトウェアの利用検知について，現在は同ソフトウェアとしての挙動を検知することに限定されており，同ソフトウェアの種別を特定するには至っていない．今後は，ファイル共有ソフトウェアの起動時におけるオーバレイネットワーク構築時のペイロード長をパラメータに加えて，SVM(Support Vector Machine) など教師付きの機械学習によって種別を特定する予定である．この場合，優れた教師データの選定が重要になると考えている．
- DoS 攻撃への防御手法について，一般的には本研究を含め，受信する攻撃パケットの数量を閾値にして防御を実施している．しかし，Web コンテンツなどの充実によって，サーバからの返信パケットが事実上サーバの負荷を高めているといえる．特に DDoS 攻撃では，1 台あたりからの受信パケットが少ない場合でも，攻撃台数が多ければサーバに深刻な影響が出る．したがって，受信パケット数ではなくサーバの負荷を閾値として，サーバ資源を保護する対応を開始し，返信パケット数の多いアクセスを攻撃と判断して，当該返信先となる送信元 IP アドレスからのパケットを逸らす手法を設計している．現在，サーバの状況を SNMP(Simple Network Management Protocol) で取得する方向で設計を進めているが，SNMP 取得する各種情報をどのように組み合わせてサーバ負荷を定義するかが重要である．併せて，攻撃の存在を判断した場合，もっとも返信パケット数の多い送信元 IP アドレスのアクセスから遮断するか，DDoS 攻撃を思われる類似の返信パケット数群を対象とするか，などの検証が必要である．
- 情報危機管理演習については，各種障害発生のシナリオに応じたサーバ設定を瞬時に適用可能な仮想環境の充実や，参加側の行動 (どのようなコマンドを実施し，どう設定を変更したかなど) を運用側が詳細に把握することで，運用の効率化を図る

ことが挙げられる．参加側に適切な対応を促すには，参加側の詳細な状況を運用側が把握する必要があるためである．このため，UNIX 系におけるシェルのヒストリ機能の拡張や，対象サーバの 2 重構成 (chroot や jail など) によって参加側の行動を把握することが考えられる．

現在では，一部のファイル共有ソフトウェアの検知や特定の DoS 攻撃に対する防御手法が多く提案されている．しかし，階層モデルにおいて，複数の階層による総合的な状況把握や対応を示した結果は非常に少なく，実際に導入して運用している事例はほとんど存在しない．これは，多くの組織でネットワーク・システムの運用管理を外部委託するケースが増える一方で，運用管理に携わる研究者がますます希有な存在となっているためである．併せて，運用管理の現場あるいはこれを忠実に再現した環境下で知見と経験を積むことが，運用管理の人材育成の基点であるにも関わらず，これを阻害する社会的・経済的なリスクが過剰に強調されていると考えられる．ハードウェアの進歩に依存するだけでなく，仕組みを理解して有効に活用できる知恵をもった人材を育てる上で，本研究が貢献できれば幸いである．

謝辞

本論文の主査である大阪府立大学　大学院工学研究科　電気・情報系専攻　知能情報工学分野　戸出英樹教授，副査である大阪府立大学　大学院工学研究科　電気・情報系専攻　知能情報工学分野　宮本貴朗教授，および大阪府立大学　大学院工学研究科　電気・情報系専攻　知能情報工学分野　吉岡理文教授には，本論文を纏めるにあたってご指導を賜り，厚く御礼申し上げます.

博士後期課程への就学を勧めてくださった，和歌山大学システム情報学センター　副センター長の内尾文隆教授，および和歌山大学システム情報学センターに在籍のまま，大阪府立大学　大学院工学研究科　電気・情報系専攻　知能情報工学分野　博士後期課程に就学を許可いただいた　和歌山大学学長　山本健慈教授ならびに　和歌山大学システム情報学センター長　河原英紀教授に厚く御礼申し上げます.

筆者の大阪府立大学　大学院工学研究科　電気・情報系専攻　知能情報工学分野　博士後期課程に在籍中，暖かくご助言くださった　大阪府立大学　大学院工学研究科　電気・情報系専攻　知能情報工学分野　谷川陽祐助教に厚く御礼申し上げます.

最後に筆者の大阪府立大学　大学院工学研究科　電気・情報系専攻　知能情報工学分野博士後期課程における研究と和歌山大学システム情報学センターにおける業務を心身両面にわたり支えてくれた妻と息子達，そして，どんな苦境でもあきらめず好奇心旺盛に取り組むよう育ててくれた両親に心から感謝いたします.

参考文献

[1] E.K.Lua, J.Crowcroft, M.Pias, “A Survey and Comparison of Peer-To-Peer Overlay Network Schemes,” IEEE Communications Surveys & Tutorials, Vol. 7, pp. 72-93, Second Quarter 2005.

[2] S.A.Theotokis, D.Spinellis, “A Survey of Peer-to-Peer Content Distribution Technologies,” ACM Computing Surveys, Vol. 36, No. 4, pp. 335-371, December 2004.

[3] 社団法人コンピュータソフトウェア著作権協会, “第 10 回「ファイル共有ソフト利用実態調査」,” http://www2.accsjp.or.jp/research/reserch11.php.

[4] 文部科学省通知資料, “個人情報の持出し等による漏えい等の防止について,” http://www.mext.go.jp/b_menu/koukai/kojin/info/002.htm#siryo1.

[5] IPA, “2011 年版 10 大脅威〜進化する攻撃…その対策で十分ですか？,” http://www.ipa.go.jp/security/vuln/documents/10threats2011.pdf.

[6] “OnePointWall,” http://www.onepointwall.jp/.

[7] “PaloAlto,” http://www.paloaltonetworks.jp/.

[8] “Skype,” http://www.skype.com.

[9] 知的財産権ワーキング・グループ等侵害対策強化事業, “ファイル共有ソフトの利用に関する調査,” http://www.meti.go.jp/meti_lib/report/2010fy01/E001204.pdf.

[10] M.Larsen, “Recommendations for Transport-Protocol Port Randomization,” RFC6056, January 2011.

[11] 元木伸宏, 泉 裕, “TCP，UDP トラヒックに着目したファイル共有検出システムの構築と運用評価,” 情報処理学会研究報告 IOT, インターネットと運用技術, Vol. 2010-IOT-8, No. 31, pp. 1-6, 2005.

[12] Microsoft TechNet, “Windows TCP/IP Ephemeral, Reserved, and Blocked Port Behavior,” http://technet.microsoft.com/en-us/library/bb878133.aspx.

[13] T.Karagiannis, A.Broido, M.Faloutsos, K.Claffy, “Transport layer identification of P2P traffic,” in Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC ’04). pp.121-134, October 2004.

[14] 藤井聖, 中村豊, 藤川和利, 砂原秀樹, “通信先ホスト数の変化に注目した異常トラヒック自動検出手法の提案と評価,” 電子情報通信学会論文誌 B, Vol. J88-B, No. 10, pp. 1922-1933, 2005.

[15] B.Claise, Ed.“Cisco Systems NetFlow Services Export Version 9,” RFC3954, October 2004.

[16] “その後の Napster 訴訟,” http://www.itlaw.jp/L&T%20-%20Napster.pdf.

[17] “LimeWire,” http://www.limewire.com/.

[18] “株式会社クロスワープ,” http://www.crosswarp.com/.

[19] “情報技術解析平成 21 年度報,” 警察庁情報通信局情報技術解析課, http://www.npa.go.jp/cyberpolice/detect/pdf/H21_nempo.pdf.

[20] 重本倫宏, 大河内一弥, 寺田真敏, “コネクション解析による P2P 通信端末検知手法,” 電子情報通信学会技術研究報告 ISEC, 情報セキュリティ, Vol.108 , No. 162, pp. 195-200, 2008.

[21] 松田崇, 中村文隆, 若原恭, 田中良明, “相互接続における順逆接続間隔を利用した P2P トラヒック弁別手法,” 電子情報通信学会技術研究報告 NS, ネットワークシステム, Vol. 106, No. 577, pp. 415-420, 2007

[22] 水谷正慶, 白畑真, 南政樹, 村井純, “複数セッションの相関関係を利用したセキュリティイベント検知手法の提案,” 情報処理学会シンポジウム論文集, Vol. 2007, No. 10, pp. 595-600, 2007

[23] 大坐畠智, 川島幸之助, “クライアント/サーバ関係に着目したピュア P2P アプリケーショントラヒック特定方式と評価,” 情報処理学会論文誌, Vol. 49, No. 2, pp. 988-998, 2008.

[24] 戸田聡, 金西計英, 矢野米雄, “トラフィックマイニングと可視化による Peer-to-Peer ファイル共有検出支援システムの構築,” 情報処理学会研究報告 DSM, 分散システム/インターネット運用技術, Vol. 2007-DSM-45, No. 38, pp. 99-104, 2007

[25] 独立行政法人情報処理推進機構, “サービス妨害攻撃の対策等調査 -報告書-,” 2010, http://www.ipa.go.jp/security/fy22/reports/isec-dos/2010_isec_dos.pdf.

[26] 西川康宏, 岡田康義, 佐藤直, “私的セキュリティポリシを利用した NGN における DoS 対策の考察,” 2009 年 暗号と情報セキュリティシンポジウム, 2E3-3, SCIS2009,

2009.

[27] 情報処理振興事業協会 セキュリティセンター, “インターネットサーバーの安全性向上策に関する調査報告書, ” 2003, http://www.ipa.go.jp/security/fy14/contents/high-availability/has.pdf.

[28] 経済産業省, “世界的規模で報告されたネット障害についての総括レポート ～Slammer ワームによる被害について～,” 2003, http://www.meti.go.jp/policy/netsecurity/030131slammerreport.pdf

[29] 金岡 晃, 岡田雅之, 勝野恭治, 岡本栄司, “DoS 攻撃経路を効率的に再構築するためのトポロジ特性を考慮した確率的パケットマーキング手法,” 情報処理学会論文誌, Vol. 52, No. 3, pp. 929-939, 2011.

[30] 村上真教, 甲斐俊文, 入江 博, 佐々木良一, “IP トレースバックにおける出国印方式の拡張と評価,” 情報処理学会論文誌, Vol. 51, No. 9, pp. 1610-1621, 2010.

[31] 泉　裕, 齋藤彰一, 上原哲太郎, 國枝義敏,. “ホストベースの DoS 攻撃防御システム SYN Packet Pacifier,” 電気学会論文誌 C, Vol. 125, No. 2, pp. 344-352, 2005.

[32] K.Lakshminarayanan, D.Adkins, A.Perrig, I.Stoica, “Taming IP packet flooding attacks,” Proc. 2nd Workshop on Hot Topics in Networks (HotNets-II), pp. 45-50, 2003.

[33] I.Stoica, D.Adkins, S.Zhuang, S.Shenker, S.Surana, “Internet Indirection Infrastructure(i3),” Proc. 2002 Conference on Applications, Technologies, Architectures, and Protocol for Computer Communications (SIGCOMM ’02), pp. 19-23, 2003.

[34] R.Ando, Z.Zhang, Y.Kadobayashi, Y.Shinoda, “A Dynamic Protection System of Web Server in Virtual Cluster Using Live Migration,” Proc. of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC ’09), pp. 95-102, 2009.

[35] 高橋朝英, 田口元貴, 小林良太郎, 加藤雅彦, “仮想計算機のリソース制御による HTTP-GET Flood 攻撃対策,” 電子情報通信学会論文誌 D, Vol. J94-D, No. 12, pp. 2058-2068, 2011.

[36] 萱島 信, 松本 勉, “プロキシを用いた分散サービス不能攻撃回避方式の提案,” 情報処理学会論文誌, Vol. 49, No. 3, pp. 1097-1104, 2008.

[37] 金森励起, 元木伸宏, 川橋 裕, 塚田晃司, “ソースアドレスルーティングによるトラフィック管理システム,” 電子情報通信学会技術研究報告 IN, 情報ネットワーク, Vol. 108, No. 342, pp. 25-30, 2008.

[38] Y.Izumi(Kawahashi), H.Tode, " Gateway Management System within the Border Based on Traffic Pattern and Source Address Routing ," Proc. of International Conference on Multimedia, Information Technology and its Applications (MITA 2009), Session C-3, pp. 75-77, 2009.

[39] FreeBSD System Manager's Manual, "ipfw," http://www.freebsd.org/cgi/man.cgi?query=ipfw.

[40] 石黒邦弘, "GNU Zebra," http://www.zebra.org/.

[41] FreeBSD System Manager's Manual, "if_bridge," http://www.freebsd.org/cgi/man.cgi?query=if_bridge.

[42] "セキュリティ総合戦略," 経済産業省情報, http://www.meti.go.jp/policy/netsecurity/downloadfiles/Strategy_Summary.pdf.

[43] "IT-KEYS," 先導的 IT スペシャリスト育成プログラム, http://it-keys.naist.jp/.

[44] "IT 危機管理演習," IT-Keys, http://it-keys.naist.jp/course/practice/it.html.

[45] "情報危機管理コンテスト," 第 12 回サイバー犯罪に関する白浜シンポジウム, http://www.sccs-jp.org/SCCS2008/sccs03.html.

[46] "インシデント体験演習," IT-KEYS, http://it-keys.naist.jp/course/practice/incident.html.

[47] "Black Hat Japan 2008 Briefings and Training - Training Session," http://www.blackhat.com/html/bh-japan-08/train-bh-jp-08-index.html.

[48] "セキュリティ・スタジアム 2004," NPO 日本ネットワークセキュリティ協会, http://www.jnsa.org/seminar/2004/seminar_20041101.html.

[49] "VMwareServer," http://www.vmware.com/jp/products/server/.

[50] "Point-to-Point Tunneling Protocol(PPTP)," RFC2637, http://www.ietf.org/rfc/rfc2637.txt, 1999.

[51] "情報危機管理コンテスト予選," 第 13 回サイバー犯罪に関する白浜シンポジウム, http://www.sccs-jp.org/SCCS2009/contest/contest.html.

[52] "情報危機管理コンテスト予選," 第 14 回サイバー犯罪に関する白浜シンポジウム, http://www.sccs-jp.org/contest2010/yosen.html.

[53] 中野宏幸, 大林厚臣, "IT 障害に関する分野横断的演習の取組み: 分野を超えた情報共有と連携協力の仕組みづくりに向けて," 社会技術研究論文集, Vol. 5, pp. 143-155, 2008.