ネットワークシステムにおける
標的型攻撃対策の研究

加藤 雅彦

システム情報工学研究科

筑波大学

２０１５年　３月

第1章 本研究の背景

1.1 インターネットの社会インフラ化

インターネットの登場により情報通信環境は劇的な変化を遂げた．遠く離れた仲間と SNS でコミュニケーションをとったり，机に座ったままで買い物を行い，同時に動画視聴を楽しんだり，クラウドにプライベートな写真を保管して仲間と共有したりするなど，我々の生活は以前とは比較にならないほど様々な情報を有効に活用できるようになった．それだけにとどまらず，電子メールや WWW はビジネスを行う上でもなくてはならないツールとなっている．もはやインターネットは，我々の日常生活を支える情報インフラとして確立された通信網と言えるだろう．

しかしながらそのような状況において，標的型攻撃と呼ばれる攻撃が増加し，重要な情報が盗まれるなどの大きな脅威となっている．標的型攻撃は境界防御などのセキュリティ設計の欠陥やアンチウイルスなどのセキュリティ対策実装の仕組みを熟知したうえで，それらの防御機能を回避して攻撃を行う．そのため，既存のネットワークシステム上で行われているセキュリティ対策では防御が非常に困難となっている．

本章ではそのような現状に至るまでの環境の変化，インターネット上の攻撃および対策の変遷を振り返るとともに，標的型攻撃の概要を解説する．

1.1.1 情報量と質の変遷

インターネットの前身となる ARPANET は米国での軍事利用を目的として作られ，その後研究を目的とした利用が始まった．日本国内においては 1990 年代初頭までは大学や研究機関のみで相互接続したネットワーク（代表的なものとしては JUNET が挙げられる）が形成され，1993 年に商用サービスが提供開始となり，一般の人々が利用可能となった[1]．商用サービス開始当初は数十 kbps 程度の速度が出る通信回線が主流[2]で，接続費も高価であったため，常時接続以外にもバッチ通信を行うといったことが普通に行われていた．流すことができる情報量は限られており，リアルタイムかつインタラクティブなアプリケーションを実現するだけの性能では無いため，限られた利用者が，電子メールやネットニュース，ファイルの送受信といったアプリケーションを使うことが一般的であった．1993 年には web ブラウザとして mosaic[3]が開発されていたが，当時の通信速度やコンピュータの能力では処理が複雑でデータ量も多く，快適な利用がでなかったため，まだ主流にはなり得なかった．

その後，通信回線の帯域増加や新たな通信サービスの登場により，通信環境の利便性は大幅に向上した．通信速度が上がることによってより高速に，大容

量の通信が可能となり，画像や文書を同時に扱うことができる Web のメリットが生きることとなった．そのため一気に一般利用が広まり，普及期を迎えて現在に至っている（図 1.1）．ISC によると 2014 年時点でのインターネット上でドメイン登録されているホスト数は 10 億ホストを超え[4]，日本国内のインターネット普及率は総務省の統計によると 2013 年末で人口普及率は 82.8%，企業では 99.9%となっている[5]．トラフィックの増加は現在でも続いており，今後のインターネットのトラフィック量について，2016 年には年間 1.1 ゼッタバイト，2018 年には年間 1.6 ゼッタバイトに達すると予測されている[6]．

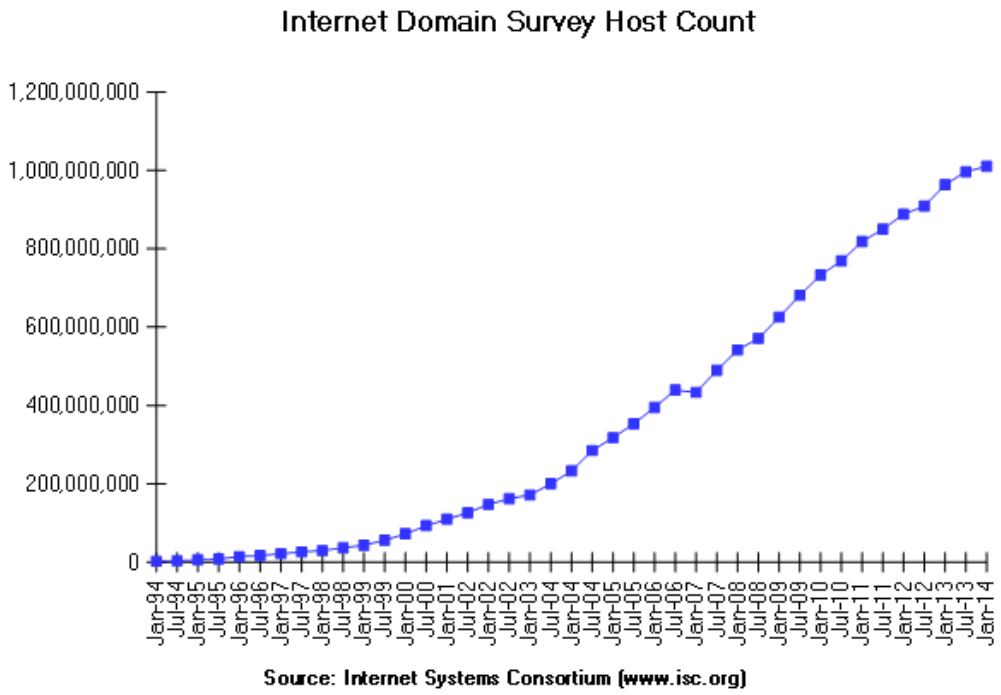

図 1.1 ドメイン登録されているホスト数の推移（ISC 資料より引用）

企業などもこの新しい通信インフラであるインターネットのメリットを生かして業務を行い，ビジネスの場として活用を進めている．初期は電子メールや公開可能な情報を広報する媒体としての役割が中心であったが，徐々に BtoB や BtoC の電子商取引などの導入が行われた．1999 年頃にはオンライントレードが登場し，インターネットを使った金融証券情報のやり取りが本格化することとなった．ネット通販なども盛んになり，2000 年には政府主導でインターネット博覧会[7]が行われ，その後首相によりメールマガジンが発行されるなど，産業，行政，学術のそれぞれの分野で盛んに活用が行われた．

現在では，インターネットでありとあらゆる情報がやり取りされている．個人情報や企業秘密情報，金融決済情報といった様々な機微情報が，インターネ

ット上を流れることは珍しいことでなく，国家機密情報までもがインターネットを通じてやりとりされるまでになった．

このように，現在のインターネットは初期とは比較にならないほど価値の高い情報が流通しており，攻撃者にとってインターネット上の情報は様々な手段を駆使し，多くの労力を割いてでも手に入れる価値のあるものとなった．

1.1.2 情報通信環境の変化

巨大な情報通信インフラとなったインターネットであるが，基本的なアーキテクチャはインターネットプロトコルを使用した LAN/WAN の相互接続と， その上で動作する TCP や UDP，さらに上位の HTTP や SMTP を使ったマルチレイヤネットワークによるクライアントサーバ型のサービスである．インターネットは中央集権的にネットワーク管理を行う国家や組織が存在せず，自律システムの集合体となっており，個々の独立した管理ポリシーを持つネットワークが相互接続することによって全体を形成している．独立したネットワークとは例えば，個人に提供されるインターネット接続サービスや，企業内ネットワーク，学術機関が運営するキャンパスネットワークなど，様々なものがあげられる．それらのネットワークを ISP が収容し，さらに ISP が IX を経由，もしくは ISP 同士が相互接続することでネットワーク全体を形成している（図 1.2）．インターネットにつながったすべての端末はデータリンク層，IP 層，TCP 層などの複数の通信レイヤを使ってバケツリレーのように，パケットが中継されていくことで通信を行っている．

初期のインターネット利用者は研究者などが中心であり，アクセス用端末はワークステーションなどに限られ，接続ノード数やサービスの種類も少なかったため，サービス提供システムも単体の機器で構成されるなど，小規模なものが中心であった．しかし利用者は爆発的に増加しており，PC やスマートフォンなど様々なデバイスを使用してインターネットに接続している．サービスも多種多様となっているため，多くのプロトコルが相互に依存しながら使用されることで非常に複雑な通信が行われている．

このようなユーザ数の増加や複雑化するサービスに対して，サービス提供者は安定してサービスを行うことができる性能および能力を持った設備を用意しなければならない．サービス提供用システムの能力を増やすためには，垂直方向の拡張，つまり単体で高機能な機器を導入するか，または水平方向の拡張，つまり，単体では性能が劣る機器を分散処理させて性能を稼ぐか，２つの方法が考えられる．垂直方向の拡張はシステム構成がシンプルで扱いやすい反面，高コスト化しやすく，柔軟な構成をとりづらい．水平分散は構成が複雑になるものの，拡張を段階的に行うことが可能でコストも抑えやすい．インターネッ

トそのものが分散システムであり，サービス提供システムにおいても水平分散システムは作りやすい．そのため，サービス提供システムは数多くのサーバをネットワークで接続して負荷を分散し，一つのネットワークシステムとしてサービスを提供するという設計が行われる（図 1.3）．

また，端末についても，グローバル IP が付与され，インターネットに直接接続を行っていたが，端末数の増加に対応できる NAT などの技術を使用してファイアウォールやルータなどでアドレス変換を行い，端末はローカル IP アドレスを使用するという構成が一般的となった．このような構成は有限であるグローバル IP アドレスを節約しつつ端末を多数接続することに一定の効果があるとともに，セキュリティ境界を設けることでネットワークを外部と内部に分割し，安全性を高めることにも役立っている（図 1.4）．

このようにサービス提供者，利用者とも，基本的な構造を変えることなく様々な工夫を行うことでアーキテクチャを変えることなく規模の増大に対応してきたが，そのことがより一層通信環境を複雑化させる要因となっているのも事実である．

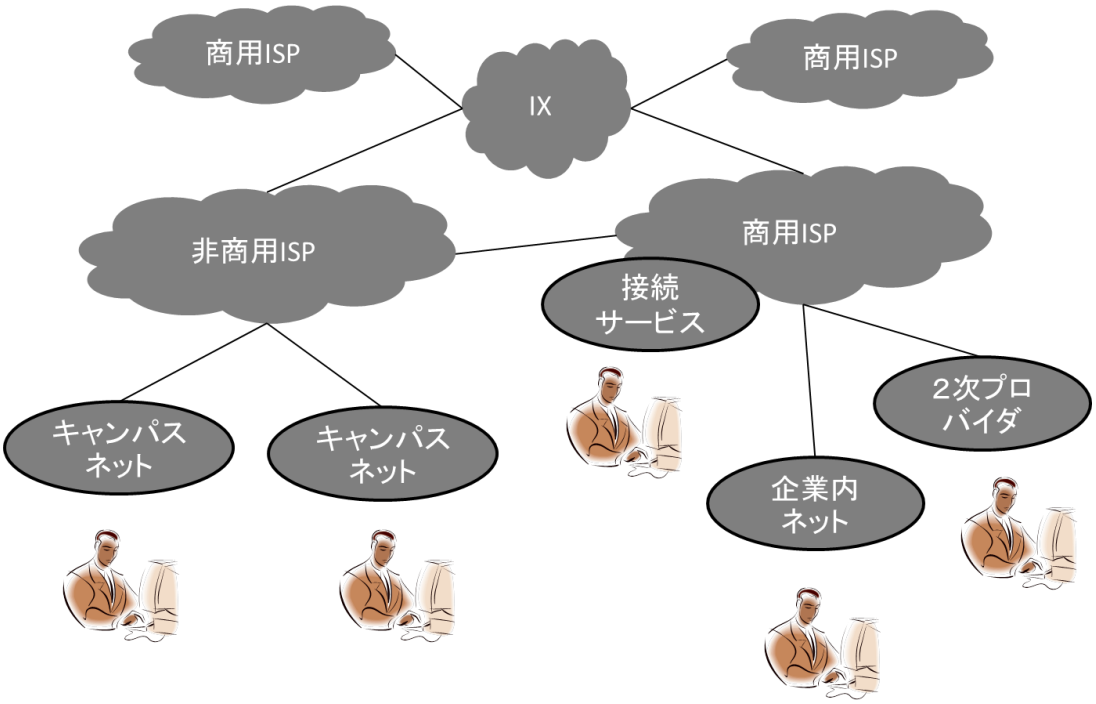

図 1.2 ネットワークの相互接続

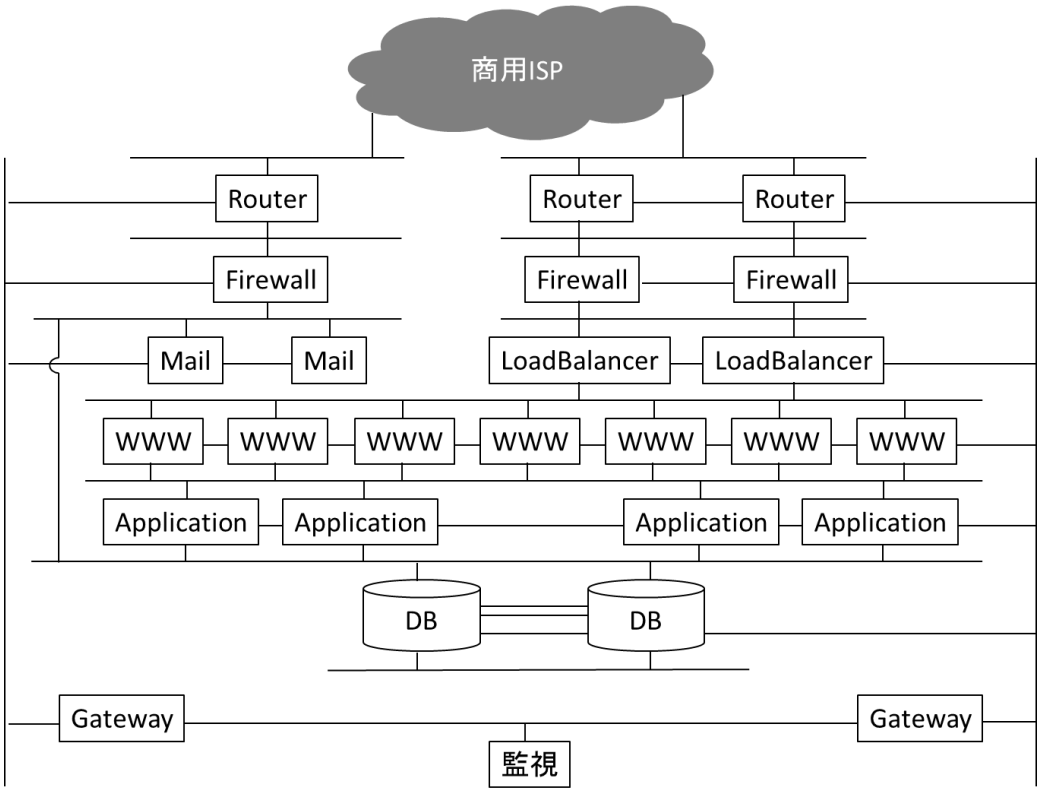

図 1.3 サービス提供用ネットワークシステム例

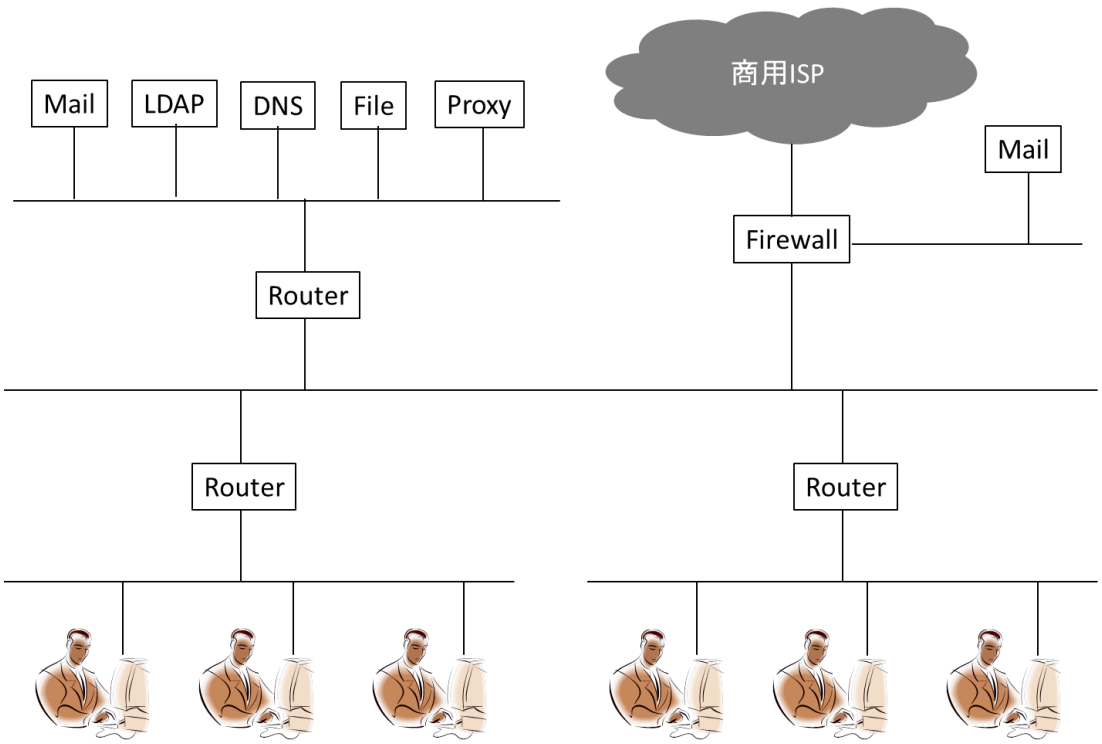

図 1.4 クライアントネットワークシステム例

1.1.3 脅威と対策の変遷

これまでの説明の通り，インターネットを取り巻く環境は激しく変化している．そして，インターネット上でおきる犯罪行為も，システムの複雑化や情報の価値に比例して，より凶悪化し，増加する傾向にある．そこで，総務省の平成 26 年版情報通信白書[8]を参考に，インターネットにおける脅威の移り変わりを概観する（図 1.5）．

インターネットに広まった最初のウイルスは 1988 年に流行したモリスワームと言われている[9]．モリスワームは単純にネットワーク上のホストに感染し続けるだけのものであったが，コンピュータネットワーク上での攻撃の存在というものを認知させた．さらに引き続いて，不正侵入による Web サイト改ざん，サーバ乗っ取りによる SPAM 送信が 2000 年初頭に増加している．その後，フィッシング詐欺やボットネットによる攻撃なども常態化していった．単純な攻撃手法では効果が薄くなるにつれ，より強力であったり，複雑であったりする攻撃手法へと遷移していることがここから読み取れる[10][11]．

攻撃者の目的も変化しており，初期は愉快犯が多かったものの，利用者が増え情報の価値が上がることで，金銭を脅し取るための DDoS 攻撃や情報窃取を目的とした攻撃がおきている．サイバーテロという言葉が用いられることからもわかるが，より深刻な犯罪行為へと移行していると考えられる．

攻撃主体も個人から組織へと移り変わっており，近年では国家の関与が疑われるなど，より組織化，大規模化していると言えるだろう．

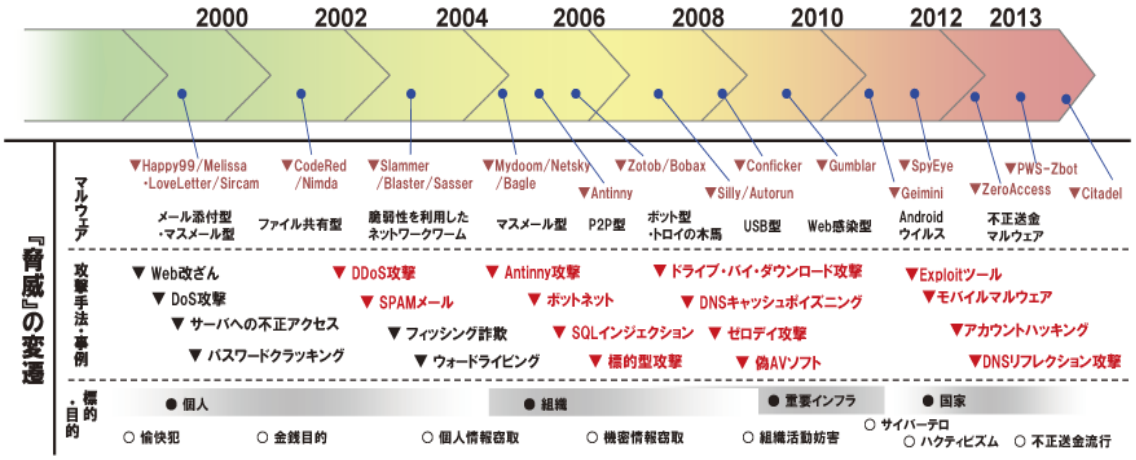

図 1.5 インターネット上の脅威の変遷（総務省 H26 情報通信白書から引用）

これらの攻撃に対して様々な対策が存在するが，基本となる対策は境界防御，つまり，ファイアウォールによるアクセス制御対策（境界防御）と，クライアント PC 保護，つまり，クライアント PC を不正プログラム感染から保護するアンチウイルス対策である．境界防御では，ファイアウォールによって，サービス提供用ネットワークシステムやクライアントネットワークをインターネットから切り離し，不要な通信のフィルタリングを行うことで攻撃から保護を行っている．アンチウイルスは，不正プログラムとして登録されたバイナリのパターンをクライアント PC 内で照合することによって検知を行い，クライアント PC に感染した不正プログラムを隔離，駆除することでシステムを攻撃から保護している．

これらの対策は，攻撃をセキュリティ境界の内部に入れることなく止めるということであり，内部ネットワークシステム構成や通信仕様がどんなに複雑化していても，既知の脆弱性に対する外部からの攻撃を防御する，という視点では合理性のある対策と考えられる．しかし，システム内に存在する情報の金銭的価値が上がり，攻撃が国家の関与であることすら想定されている事件も存在することが示すように，攻撃者のモチベーションは非常に高まっている．外部からの攻撃や情報窃取が困難であれば，攻撃者は攻撃対象に特化した攻撃手法を用い，ネットワークシステム内部への侵入を試み，内部から重要な情報を入手しようとするであろうことは想像に難くない．

このような背景から，既存の対策を迂回し，より巧妙かつ複雑な手法で重要な情報やシステムを狙う標的型攻撃が登場した．標的型攻撃はソフトウェアが持つ脆弱性への攻撃などと異なり，過去から大きな変化がないネットワークシステムのセキュリティ設計手法を悪用して攻撃が行われており，今までの安全対策手法の前提を覆すものである．次節ではこの標的型攻撃について詳細な説明を行う．

1.2 標的型攻撃

標的型攻撃とは，ある特定の組織や人物を標的として，ソーシャルな手段を利用しつつ複数の段階を経て，様々な攻撃手法を組み合わせ，持続的に組織内ネットワークに侵入し，最終的に情報の窃取やシステム破壊などを行う攻撃である．標的型攻撃は攻撃の全体像を指す言葉であり，ある特定の技術的な攻撃手法を表すものではないことに注意が必要である．標的型攻撃の具体的な内容として，アンチウイルスベンダー等から様々な定義や解説が公開されている[12][13]が，本論文では日本国内を標的とした事例が幅広く取り上げられている IPA による定義を前提とすることとした．

1.2.1 標的型攻撃の事例

標的型攻撃が最初に観測されたのは 2006 年頃であるが，2010 年に Google 社が攻撃の事実を公表したことで一気に注目されるようになった．ここでは実際に起きている標的型攻撃の事例を表 1.1 に示す．標的型メールの利用が多いが，USB による侵入や正規アカウントの利用，Web サーバへの侵入など，標的の環境に合わせた攻撃の多様性が事例から見て取れる[14].

警視庁の調査では，2014 年上期だけでも国内事例として標的型メール攻撃と呼ばれる攻撃が 216 件発生したとされている[15]．秘密情報の流出は避けられたとしても重要な情報を狙った攻撃は今現在でも止まることなく続いており[16][17]，その対策が急務となっている．

報道	標的	攻撃目的	攻撃の概要・特徴
2010/1	Google 社など多数	秘密情報の窃取	Operation Aurora と呼ばれる攻撃．Web アクセスにより組織内部の PC が感染し，Web メールのアカウント情報等が窃取された[18].
2010/7	某国の核燃料処理施設	装置の誤作動	Stuxnet と呼ばれる攻撃．USB メモリから組織内部に侵入し，組み込み機器（PLC）の開発環境に感染．開発環境から組み込み機器にコードを書き込む際に悪意のあるコードが仕込まれ，PLC を誤作動させることで装置の正常動作を妨害した[19].
2011/2	エネルギー業界企業	秘密情報の窃取	Night Dragon と呼ばれる攻撃．Web サーバへの侵入や標的型メールなど複数の侵入手段が使われ，組織内部のアカウント情報を使って重要情報が窃取された[20].
2011/4	RSA 社	秘密情報の窃取	標的型メールが攻撃に使用され，遠隔操作によって組織内ネットワークで持続的に攻撃が行われ，秘密鍵に関する情報が漏えいした [21].
2011/9	三菱重工業など多数	秘密情報の窃取	最初に関連業界団体の PC へ侵入が行われ，次に業界団体からの連絡を偽装した標的型メールが本当の標的に対して送られた．偽装メールを開くことで標的となる組織内 PC に不正プログラムが感染し，遠隔操作によって軍事関連情報が狙われた[22].

2011/10	衆議院	秘密情報の窃取	標的型メールが使用され，議員の PC が攻撃対象となり，ID やパスワードが窃取された[23]．なお，同様の攻撃が参議院に対しても行われている．
2012/1	宇宙航空研究開発機構	秘密情報の窃取	震災関連のなりすましメールが使用され，組織内の PC に不正プログラムが感染．感染 PC から約 20 か月間不正な通信が発生していた[24]．
2013/3	韓国の放送局，銀行など	システムの破壊	攻撃には標的型メールが使用され，組織内の PC が感染．そこからパッチ配布サーバに不正プログラムが埋め込まれた．パッチ適用した PC でディスクを破壊するプログラムが動作して起動不能となることで，大規模なシステム障害に発展した [25]．
2014/1	日本原子力研究開発機構	秘密情報の窃取	動画再生ソフトウェアの更新を行うことで，組織内の PC が感染．感染 PC から外部のサイトに不正な通信が発生していた[26]．

表 1.1 標的型攻撃の事例

1.2.2 標的型攻撃の流れ

IPA によると，標的型攻撃は計画立案，攻撃準備，初期潜入，基盤構築，内部侵入・調査，目的遂行，再侵入の 7 段階で攻撃が行われるとされている[27]．ここでは具体的にネットワークシステムに侵入する第 2 段階から第 3 段階までを図 1.6 で，ネットワークシステム侵入後に目的を達成するまでの第 4 段階から第 6 段階を図 1.7 で説明する．

第 1 段階（計画立案）：

攻撃者は攻撃目標を設定し，攻撃対象の調査を行う．インターネット上で公開されている情報や SNS の情報などを利用し，標的に関連する情報を収集する．

第 2 段階（攻撃準備）：

C&C サーバや踏み台の準備を行う．そのために，標的と関連のある別組織などへの攻撃を行う．（図 1.6（1））

第 3 段階（初期潜入）：

第 2 段階で侵入した端末を踏み台とし，標的型メールなどを送付し，標的となる組織や個人の端末を不正プログラムに感染させる（図 1.6（2））．標的型

メールは送信者が成りすまされていたり，標的となる人物と関係のある内容が記載され，メールを読んだ人物が本文中の URL をクリックしやすくなっていたり，ファイル名やアイコンが偽装され，添付ファイルを開きやすくなっていたりするよう，様々な工夫がされている．不正プログラムに端末が感染すると，C&C サーバに向かって感染を知らせる通信が発生する（図 1.6 (3)）．

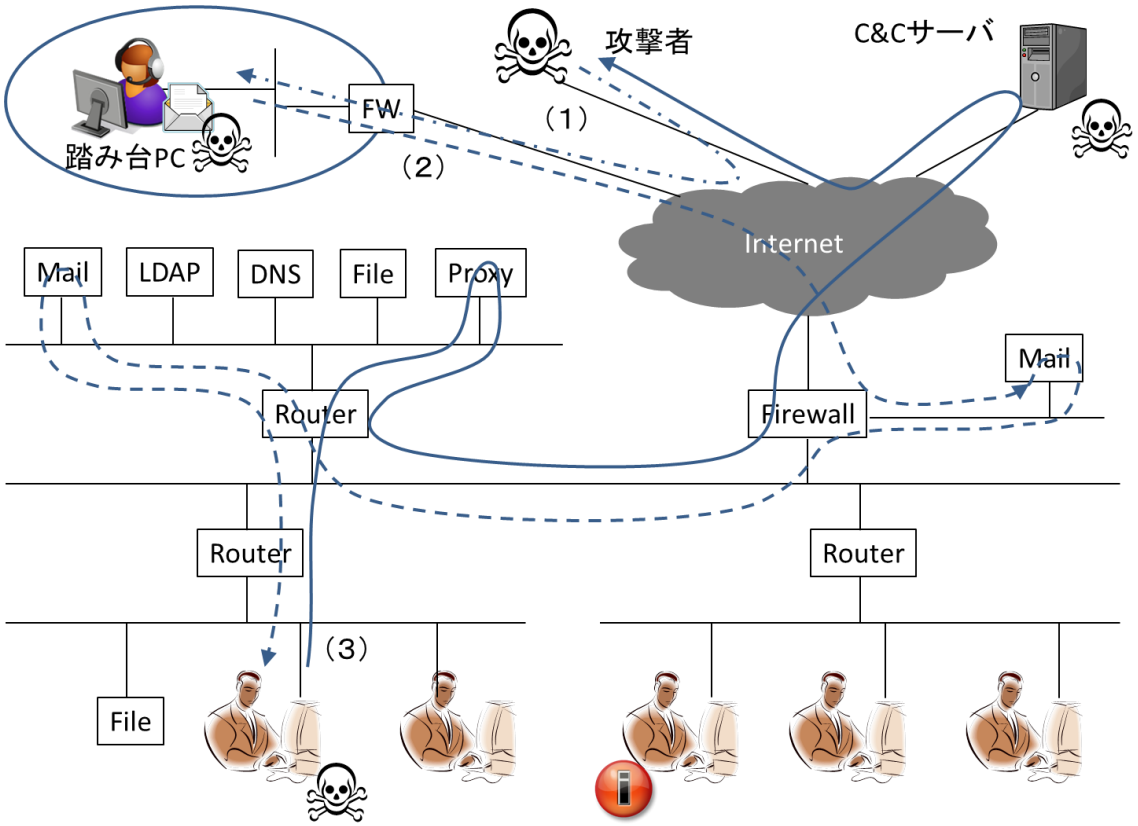

図 1.6 標的型攻撃の流れ（第 2～3 段階）

第 4 段階（基盤構築）：

侵入に成功した端末の不正プログラムは新たなプログラムをダウンロードし，バックドアを開設する．攻撃者はそのバックドアを使って端末を遠隔操作し，標的となるネットワークシステム構成情報やアカウント情報を収集する（図 1.7 (1)）．バックドアは内部ネットワークで使われている通信プロトコルを使用するなどして，通信が検出されないようにする．

第 5 段階（内部侵入・調査）：

攻撃者はさらに，バックドアを経由し，内部ネットワークの他端末やサーバへ侵入し，バックドアを増やすとともに目的の情報を探す（図 1.7 (2)）．

第 6 段階（目的遂行）：

目的とする情報を見つけると，情報を断片化して複数の端末から分割送信するなどして，攻撃を検出されないようにしながら窃取を行う（図 1.7 (3)）．

第 7 段階（再侵入）：

開設したバックドアを使用して再びネットワークシステム内で活動を行う．

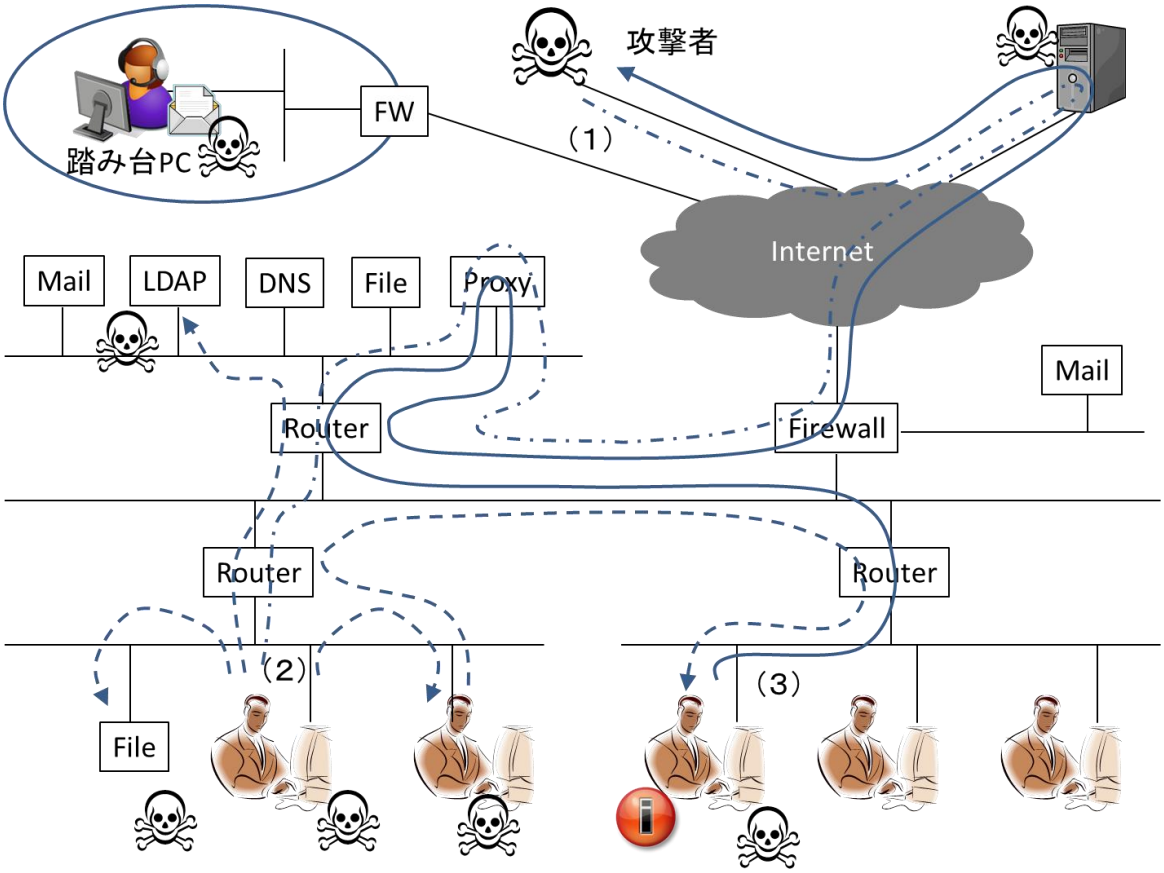

図 1.7 標的型攻撃の流れ（第 4～6 段階）

このように，単純に不正プログラムが動作するというだけではなく，遠隔操作により最終目的に到達するまで，様々な攻撃手段を利用し，複数の段階を経て攻撃が持続的に行われる．これが標的型攻撃の流れである．

次に侵入後の活動に利用される遠隔操作通信の仕様について述べる．IPA によると第 4 段階以降で使用される遠隔操作の通信プロトコルは図 1.8 にあるように，感染端末から外部ネットワークに直接通信を行うものが約半数，Web アクセスに利用される HTTP などのプロトコルを使用して通信を行うものが残り半数存在し，その一部は HTTP プロキシサーバに対応している[28]．たとえファイアウォールがあったとしても，内部ネットワークから外部ネットワークへ出

ていく不要な通信を制限していなければ，これらの通信はそのままファイアウォールを通過してしまい攻撃が成立する．内部ネットワークから外部ネットワークへ出ていく不要な通信が制限されていたとしても，内部ネットワークから外部ネットワークへ出ていくために必要な通信，例えば HTTP などを偽装した場合には，通信プロトコルとして問題がなければやはりファイアウォールを通過してしまい攻撃が成立する．このように，正規に利用する通信プロトコルや通信経路などを偽装し，悪用することで遠隔操作通信の検出をより困難なものとしている．

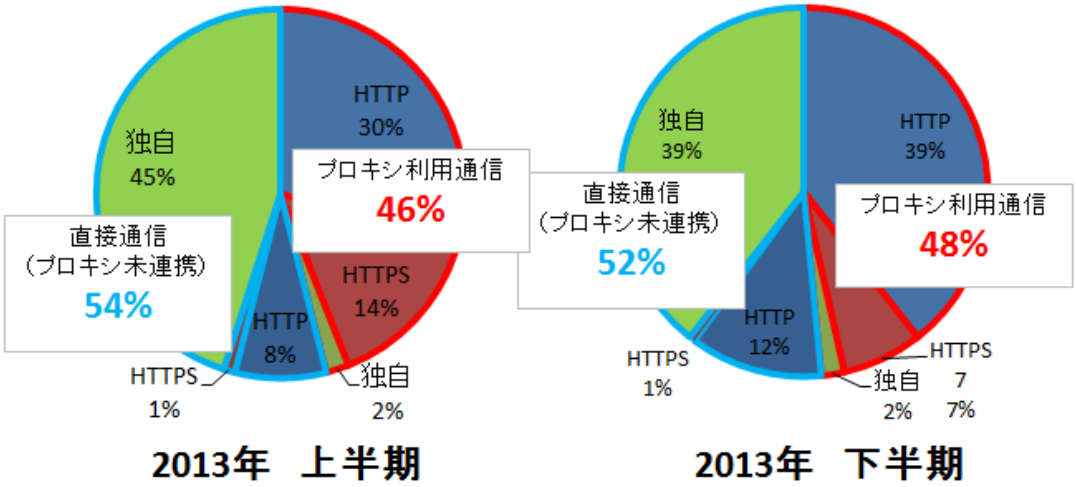

図 1.8 遠隔操作の通信方法（IPA 資料より引用）

1.2.3 標的型攻撃の特徴

以上，標的型攻撃の定義や実際の事例を踏まえ，あらためてネットワークシステム防御の視点から見た標的型攻撃の特徴をまとめると下記のとおりと考えられる（図 1.9）．

1）様々な手段を用いて内部ネットワークに侵入し，侵入行為および内部ネットワークの探索行為を持続的に続ける

2）侵入に成功した端末を攻撃者が遠隔操作することにより，目的を達成するまで複数の手法を用いて段階的に攻撃を行う

3）ネットワークシステムの提供機能として必要な通信プロトコルを偽装することで攻撃の隠ぺいを行う

これらの特徴を考慮しつつ，次章では具体的に本研究で想定する，攻撃対象となるネットワークシステムおよびそれらを守る既存の防御手法の解説を行い，

標的型攻撃対策の方向性を議論する.

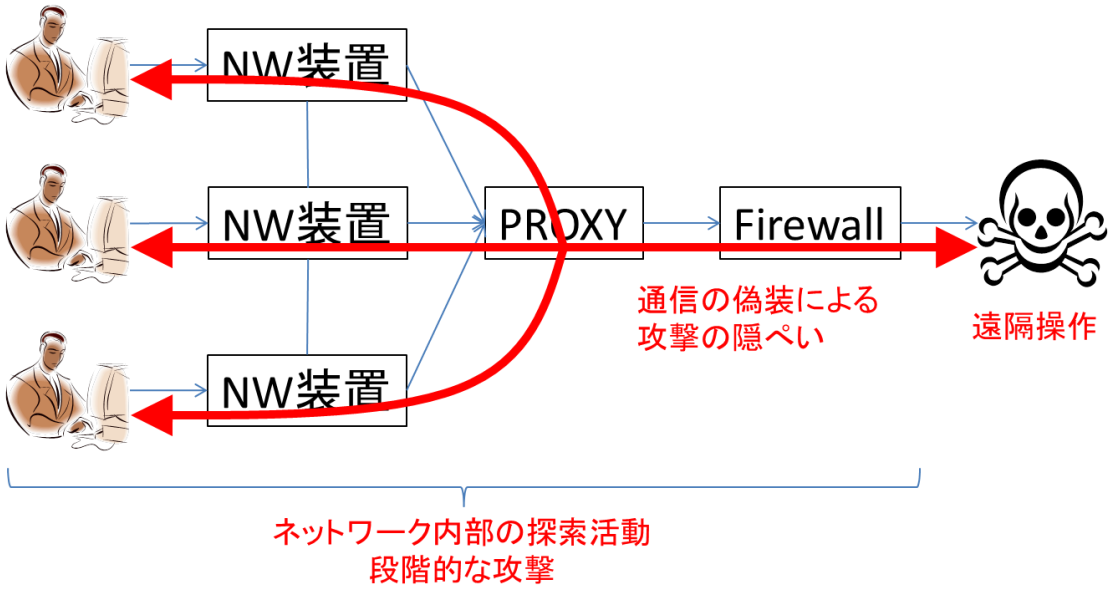

図 1.9 標的型攻撃の特徴

参考文献

［1］日本ネットワークインフォメーションセンター：インターネット歴史年表，日本ネットワークインフォメーションセンター（オンライン），入手先 <https://www.nic.ad.jp/timeline/index.html>（参照 2014-10-29）
［2］NTT:電気通信年表，NTT（オンライン），入手先 <http://www.hct.ecl.ntt.co.jp/exhibitions/chronology/>（参照 2014-10-29）
［3］University of Illinois: NCSA Mosaic, University of Illinois (online), available from <http://www.ncsa.illinois.edu/enabling/mosaic> (accessed 2014-10-29)
［4］Internet Systems Consortium: ISC Domain Survey, Internet Systems Consortium (online), available from <http://www.isc.org/services/survey/> (accessed 2014-10-29)
［5］総務省：平成 25 年通信利用動向調査の結果，総務省（オンライン），入手先 <http://www.soumu.go.jp/johotsusintokei/statistics/data/140627_1.pdf>（参照 2014-10-29）
［6］Cisco Systems: The Zettabyte Era-Trends and Analysis, Cisco Systems, Inc. (online), available from <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.html> (accessed 2014-10-29)
［7］総務省：平成 13 年版 情報通信白書，総務省（オンライン），入手先 <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h13/html/D3013000.htm>（参照 2014-10-29）
［8］総務省：平成 26 年版 情報通信白書，総務省（オンライン），入手先 <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h26/pdf/26honpen.pdf>（参照 2014-12-27）
［9］JPCERT コーディネーションセンター：インターネットセキュリティの歴史 第 1 回 「Morris ワーム事件」，JPCERT コーディネーションセンター（オンライン），入手先 <https://www.jpcert.or.jp/tips/2007/wr071202.html>（参照 2014-10-29）
［１０］総務省：平成 26 年版情報通信白書，総務省（オンライン），入手先 <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h26/html/nc143210.html>（参照 2014-10-29）
［１１］JPCERT コーディネーションセンター：JPCERT/CC セキュリティインシデント年表，JPCERT コーディネーションセンター（オンライン），入手先 <http://www.jpcert.or.jp/magazine/chronology/>（参照 2014-10-29）
［１２］トレンドマイクロ株式会社：「持続的標的型攻撃」とは？，トレンドマイクロ株式会社（ オンライン），入手先 <http://about-threats.trendmicro.com/relatedthreats.aspx?language=jp&name=Understanding%20Highly%20Targeted%20Attacks>（参照 2014-10-29）.

［１３］株式会社カスペルスキー：Advanced Persistent Threat（APT）攻撃：今までにない高度なマルウェア，Kaspersky Labs Japan（オンライン），入手先〈http://www.kaspersky.co.jp/downloads/pdf/advanced-persistent-threats-not-your-average-malware_kaspersky-endpoint-control-white-paper_jp.pdf〉（参照 2014-10-29）．
［１４］警察庁警備企画課・情報技術解析課：平成 25 年上半期のサイバー攻撃情勢について，警察庁（オンライン），入手先〈http://www.npa.go.jp/keibi/biki3/250822kouhou.pdf〉（参照 2014-12-11）．
［１５］警察庁：平成 26 年上半期のサイバー空間をめぐる脅威の情勢について，警察庁（オンライン），入手先〈https://www.npa.go.jp/kanbou/cybersecurity/H26_kami_jousei.pdf〉（参照 2014-10-29）
［１６］農林水産省：農林水産省へのサイバー攻撃に関する調査結果（中間報告）の公表について，農林水産省（オンライン），入手先〈http://www.maff.go.jp/j/press/kanbo/hisyo/130524_1.html〉（参照 2014-12-11）．
［１７］外務省：外務省ネットワークから外部への情報流出，外務省（オンライン），入手先〈http://www.mofa.go.jp/mofaj/press/release/25/2/0205_07.html〉（参照 2014-12-11）．
［１８］Google: A new approach to China, Google Inc. (online), available from〈http://googleblog.blogspot.jp/2010/01/new-approach-to-china.html〉(accessed 2014-12-7).
［１９］David Kushner: The Real Story of Stuxnet, IEEE Spectrum (online), available from〈http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet〉(accessed 2014-10-29)
［２０］McAfee：世界のエネルギー産業を狙うサイバー攻撃:“Night Dragon”，McAfee Inc.（オンライン），入手先〈http://www.mcafee.com/japan/security/report/download.asp?no=56〉(参照 2014-12-7)
［２１］RSA FraudAction Research Labs: Anatomy of an Attack, EMC Corporation (online), available from〈https://blogs.rsa.com/anatomy-of-an-attack/〉(accessed 2014-12-7)
［２２］三菱重工業株式会社：コンピューターウイルス感染に関する調査状況について（その 1），三菱重工業株式会社（オンライン），入手先〈http://www.mhi.co.jp/notice/notice_110930.html〉（参照 2014-12-7）．
［２３］衆議院：衆議院へのサイバー攻撃報道に関する件，衆議院（オンライン），入手先〈http://www.shugiin.go.jp/internet/itdb_kaigiroku.nsf/html/kaigiroku/009017920111114005.htm〉（参照 2014-12-7）．
［２４］宇宙航空研究開発機構：JAXA におけるコンピュータウイルス感染の発生及び情報漏洩の可能性について，宇宙航空研究開発機構（オンライン），入手先〈http://www.jaxa.jp/press/2012/11/20121130_security_j.html〉（参照 2014-12-11）．

［２５］染谷征良: マスターブートレコードを破壊する韓国企業へのサイバー攻撃，その全体像と教訓とは？，Trend Micro Inc.（オンライン），入手先〈http://blog.trendmicro.co.jp/archives/6923〉（参照 2014-10-29）
［２６］日本原子力研究開発機構：コンピュータウイルス感染による情報漏えいの可能性について，日本原子力研究開発機構（オンライン），入手先〈http://www.jaea.go.jp/02/press2013/p14010601/index.html〉（参照2014-10-29）
［２７］情報処理推進機構：「標的型メール攻撃」対策に向けたシステム設計ガイド，情報処理推進機構（オンライン），入手先〈http://www.ipa.go.jp/files/000033897.pdf〉（参照 2014-10-29）
［２８］情報処理推進機構：「『高度標的型攻撃』対策に向けたシステム設計ガイド」の公開，情報処理推進機構（オンライン），入手先〈https://www.ipa.go.jp/files/000042039.pdf〉（参照 2014-10-29）

第2章 従来対策とその課題

2.1 ネットワークシステムとは

先に述べたように，インターネットは個々の独立したネットワークが相互接続することによって形成されている．本研究では個々のネットワークはファイアウォールなどによってインターネットと接続され，アクセス制御されているという前提とし，保護対象となるネットワークを内部ネットワーク，そうでないネットワークを外部ネットワークとしている．しかし，現実に利用されているネットワークシステムは目的や規模によって千差万別である．また，詳細な内部構成が公にされることもない．そこで本研究では，以下 2.1.1 項および 2.1.2 項で示すようなネットワークシステムを攻撃対象として想定し，標的型攻撃対策を議論することとした．

2.1.1 イントラネット

本研究で想定する，企業内で使用される情報インフラ基盤で，各種の業務用サーバやクライアント PC などが接続されている．実際の環境においてその規模は様々であるが，本研究ではクライアント PC が数台から数百台の場合を想定する（図 2.1）．部署やビルのフロア単位などで，ルータを使用してサブネット分割し（図 2.1 (1)），各クライアント PC はスイッチングハブに接続される（図 2.1 (2)）．各サブネットはセンターの L3 スイッチに収容され相互に接続される（図 2.1 (3)）．クライアント PC については DHCP などを使用してプライベート IP アドレスが自動的に割り振られるが，そのアドレスは管理されているものとし，確認可能とする．

業務用のサーバについては次のものを想定する．サーバのアドレスもすべてプライベート IP アドレスを使用し，DNS による名前解決が可能とする．部署で用意したファイルサーバ(File)は部署内ネットワークに存在し（図 2.1 (4)），部署間で共通に使用されるサーバについては，専用のサーバセグメントに収容される（図 2.1 (5)）．サーバは自ドメイン及び他ドメインとの電子メール送受信を行うメールサーバ(Mail)，ユーザ認証を行うためのディレクトリサーバ(LDAP)，自ドメイン及び他ドメインの名前解決を行うネームサーバ(DNS)，ファイルストレージとしてのファイルサーバ(File)，内部ネットワークからインターネットへアクセスするためのプロキシサーバ(Proxy)などが存在する．

インターネットとの接続においては，ファイアウォールを設置し（図 2.1 (6)），外部ネットワークからの通信を制限する．内部ネットワークでプライベート IP アドレスを使用するため，外部ネットワークへの通信はファイアウォールでアドレス変換を行う．また，DMZ にはメールを中継するためのメール

リレーサーバ(Mail)が設置されているとする（図 2.1 (7)）．電子メールについてはプロトコルの仕様上，外部ネットワークから内部ネットワークへの通信が必要となるため，メールリレーサーバに対して， TCP port 25 のインバウンドトラフィックをファイアウォールで開放する．内部ネットワークのメールサーバは DMZ のメールリレーサーバを経由してメールの送受信を行う．それ以外の外部ネットワークからの通信についてはファイアウォールで拒否する．内部ネットワークから外部ネットワークへの通信については制限しない．

図 2.1 に，通信の例としてクライアント PC のブラウザでプロキシサーバが指定されている場合に Web 閲覧を行うときの HTTP 通信経路を赤い矢印で示し，電子メールの送信を行う時の SMTP 通信経路を青い矢印で示す．

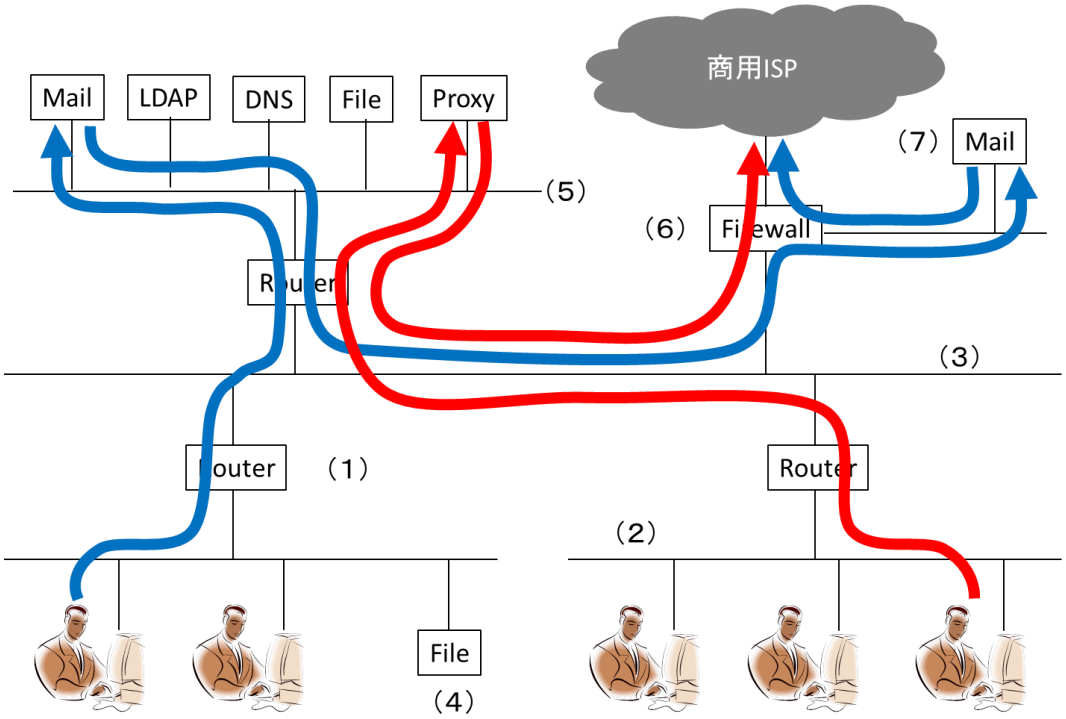

図 2.1 イントラネットの例

2.1.2 Web サービス提供用システム

インターネットで提供されるユーザ登録型の Web サービスで，同時数百セッション規模の接続に耐えるように構成されていると仮定する（図 2.2）．インターネットとの接続部分については，回線接続用のルータ（図 2.2 (1)），外部ネットワークからの攻撃を防御するためのファイアウォール（図 2.2 (2)），負荷分散のためのロードバランサ（図 2.2 (3)）で構成し，すべてを Active-Standby 型の冗長構成とし，機器の故障が発生した際にも単一障害であればサービス提供が停止しないようにしている．Web サーバ(WWW)は水平分散され，負

荷に応じて増強される（図 2.2 (4)）．登録ユーザがログインしてアクセスするためにユーザのトランザクション管理を行っており，ユーザ別の画面生成用にアプリケーションサーバ(Application)が Web サーバと別のセグメントで接続されている（図 2.2 (5)）．さらに，アプリケーションサーバの裏側に二重化されたデータベースサーバ(DB)が配置されている（図 2.2 (6)）．データベースはデータのロストを回避するために同期をとりながら Active-Standby 型で冗長化されている．

また，ユーザへ情報をプッシュするためにメールサーバ(Mail)が別途設置されている（図 2.2 (7)）． Web サービスへのアクセスとトラフィックを分けるため ISP との接続を別途用意している（図 2.2 (8)）．同じメールサーバ上で登録ユーザに合わせたメール内容を生成するアプリケーションが動作しており，裏側のセグメントからユーザ情報が登録されているデータベースへアクセスしている（図 2.2 (9)）．このネットワークシステムに接続されているすべての機器が監視専用のセグメントに接続されており（図 2.2 (10)），24 時間稼働状況が監視されている．

図 2.2 に通信の例として登録ユーザがログインして Web の閲覧を行うときに必要となる通信経路を赤い矢印で，システムが電子メールを送信するときに必要となる通信経路を青い矢印で示す．

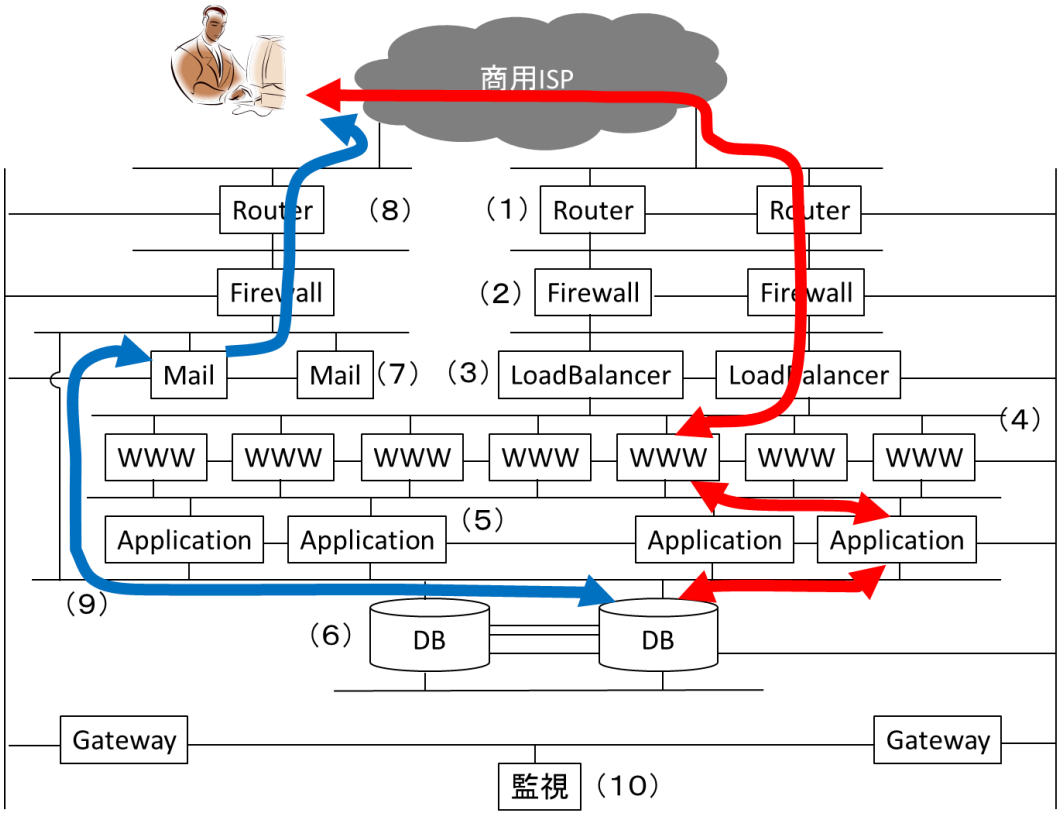

図 2.2 Web サービス提供システムの例

2.2 ネットワークシステムに対する攻撃

対策を検討するにあたって，このようなネットワークシステムに対して適用される，従来型の攻撃行為について概説する．これらのネットワークシステムに対する攻撃は数多く存在するが，ここでは今までに行われている代表的な攻撃行為および攻撃事例を示す．

2.2.1 不正侵入

境界防御を抜けて内部ネットワークに侵入する行為である．イントラネット，サーバシステムともに攻撃対象となり得る．その手法は数多く存在し，サーバなどに使用されるソフトウェアの脆弱性を利用して侵入する，境界防御の外から総当たり，もしくはアカウントリストを使いログインを試行して侵入する，テスト用などに使用されているために管理が杜撰なサーバの設定不備を利用して侵入する，などがある．不正侵入後，標的型攻撃であればさらに重要な情報を窃取するといった行為が行われるが，その他にも SPAM 送信用の踏み台として使う，Botnet に組み込んで必要な時に使う，フィッシングサイトを構築する，など，多種多様な悪用が行われる．不正侵入については現在でも日常的に起きており， SSH を使用して不正ログインされ，踏み台として利用されたなどの事例が数多く報告されている[1]．

2.2.2 Web 改ざん

不正な手段によって Web 上の情報を書き変える攻撃である．サーバシステムが攻撃対象となる．Web アプリケーションの実装上の欠陥をついてリモートでコマンドを実行させてコンテンツを書き換える，といった攻撃が行われたり，サーバの設定不備によってアクセス制御が正しく行われていない場合などは，管理インターフェースから直接ファイルを置き換えられたりすることもある．

事例としては 2009 の gumblr ウイルスによる Web 改ざん事件が挙げられる．悪意のあるスクリプトを埋め込まれた Web サイトを閲覧することでクライアントがウイルス感染し，感染したクライアントで使用される Web サイト管理アカウントが窃取され，窃取したアカウントを利用して Web サイトが書き変えられるといったことが多発した[2]． 2013 年 9 月には警察庁からの注意喚起が出る[3]など，現在でも継続的に Web 改ざんが行われている状況にある．

2.2.3 DDoS

主にサーバシステムが攻撃対象となる．サービスの提供を目的とするネットワークシステムがインターネットと接続する際には，ISP との接続回線，接続

用ルータ，ファイアウォールといった機器を使用する．攻撃者はこれらのリソースを枯渇させ，サービスが使用できなくなることを目的とした攻撃を行う．具体的には，大量のパケットを送出して接続回線の容量を使い切る ICMP flood や UDP flood，大量のコネクションを張ることでファイアウォールやサーバの接続管理用テーブルを溢れさせる syn flood や HTTP flood などがある．攻撃手法は複数存在し，専用の攻撃ツールを使用する場合や，ブラウザのリロードボタンを押し続ける F5 攻撃といったものがある．戻りパケットを必要としない攻撃はソース IP を偽装しても攻撃が可能であり，その場合は攻撃元の検出が困難となる可能性がある． TCP 接続を行うような攻撃は戻りパケットを必要とする攻撃手法を用いると，攻撃元の IP アドレス特定が可能な場合もあり，ファイアウォールでソース IP を遮断したり，上流 ISP で攻撃元を遮断したりするといった対処が行いやすい． botnet を使用しての DDoS 攻撃，また，ソースアドレスを攻撃対象そのものに偽装して，DNS サーバや NTP サーバなどに対して，クエリのデータ量が小さく，レスポンスのデータ量が大きいリクエストを発行する DRDoS といった手法も近年では多く利用されている．

事例としては，2005 年に，首相官邸のホームページに対して行われた DDoS 攻撃が挙げられる[4]．この攻撃は，攻撃手法を変化させながら数か月にわたって続き，Web サイトへのアクセスを困難とするなど，大きな被害を出した．また，近年では SPAM 対策関連組織に対して最大 300Gbps の DDoS 攻撃が行われ，ネットワーク設備の処理能力を超えて他のネットワークに遅延が発生するなどの影響が起きている[5]．

2.3 ネットワークシステムの防御

前節のような攻撃に対して，用いられている既存の防御策としては次のようなものがある．なお，それぞれ製品としては複数の機能を持っているものも多く，一意に分類が可能ではないことが多いため，すべてが以下のカテゴリに分類できるというものではない．

2.3.1 ファイアウォール

ネットワークをセグメントに分割し，アクセスコントロールを行う．ソース IP アドレス（グループ），デスティネーション IP アドレス（グループ），TCP/UDP/ICMP 等のプロトコル（グループ），ポート番号（グループ）などを組み合わせてアクセス制御ルールを定義し，そのルールに合致するパケットに対して通過，拒否，破棄などの処理を行う．また，処理時には設定に合わせてログを取得することも可能である．どの通信レイヤのアクセス制御ができるかで種類が分かれており，IP レイヤで制御が可能な場合はパケットフィルタリング，

IP とトランスポートレイヤで制御可能な場合はサーキットゲートウェイ，さらに上位層も含んで制御可能な場合はアプリケーションゲートウェイとされるが，現在ではまとめてファイアウォールと呼称されることが多い．

ネットワークアドレス変換（NAT や NAPT）機能を併せ持つ場合は，パケット通過時にアドレス変換を行う．これにより内部ネットワークでプライベート IP アドレス，外部ネットワークでグローバル IP アドレスを割り振り，通信の中継を行うことが可能である．境界防御を行うための基本的なデバイスである．パケットのヘッダによる制御であり中身は見ておらず，通過時に通信プロトコルのチェックのみを行っているため，ルール上，通過可能として定義された通信はプロトコル的に問題がなければ，データの内容にかかわらずそのまま通過する．実装としてはソフトウェア，ハードウェアのどちらも存在し，非商用のものは iptables[6]，pfSense[7]，商用としては checkpoint 社の Firewall-1[8]，Juniper 社の SSG シリーズ[9]などがあげられる．

2.3.2 IDS/IPS

IDS は Intrusion Detection System の略である．ネットワークに流れる通信をタップし，攻撃と定義された通信のビットパターンや通信量をチェックすることでアラートを出す． IPS は Intrusion Prevention System の略で，上記の動作に加えて，インラインで動作させることでアラートだけではなく通信の遮断を行う．IDS 同様，攻撃と定義されるパターンは事前に定義されているため，パターンが存在しない攻撃は検出されない．逆に，パターンに一致すれば正常な通信を誤検知するといったこともありうる．実装としてはソフトウェア，ハードウェアのどちらも存在し，非商用のものは snort[10]，商用としては IBM 社の Security Network IPS シリーズ[11]などがあげられる．その他に， OS 上のファイル変更などを監視するソフトウェアを対象機器にインストールするタイプの IDS もあるが，それはホスト型 IDS と呼ばれて区別されている．

2.3.3 AntiVirus

ソフトウェアを対象機器にインストールし，事前に定義されたビットパターンを元にして不正なプログラムを検出，駆除する．プログラムの動作をトレースし，その振る舞いによって不正なプログラムかどうかを判断する，ふるまい検知型も存在する．クライアント PC にインストールして使用するものが主流であるが，企業ではゲートウェイ型も使われ，電子メールの添付ファイルなどに対してユーザに配信する前にウイルスチェックを行うものもある．攻撃検知パターンはアンチウイルスベンダーが不正プログラムを収集・分析することで生成されるため，収集できていない不正プログラムはパターンが存在せず，検

知，駆除ができないという課題がある．標的型攻撃では標的専用の不正プログラムが開発されて使用されることも多く，その場合は当該不正プログラムの流通範囲が狭い．そのため検知パターンが作成されずに検出されないという問題が発生しやすい．非課金のものとしては Microsoft Security Essentials[12]などがあり，商用としては Trendmicro 社，Symantec 社，Kaspersky 社などのアンチウイルス製品[13][14][15]が挙げられる．

2.3.4 AntiDDoS

主にネットワークリソースを浪費するタイプのサービス妨害攻撃を防御する．インターネットサービスプロバイダ側に装置を設置するものと，利用者側のネットワークに装置を設置するものがある．基本的には，大量に到着するトラフィックを高速に破棄するという動作を行うが，BGP などを使ってネットワークの経路を操作するものもある．利用者側に装置を設置する場合はプロバイダと利用者間の通信帯域を埋められた場合に対処が困難であるため，サーバリソース消費型（TCP のコネクションテーブルをあふれさせるような Syn Flood や HTTP Flood といった攻撃）防御に適している．プロバイダとの通信帯域を埋められた場合にはプロバイダ側での対処が必要となる．商用サービスとして NTT 社や IIJ 社がサービス提供[16][17]している．

2.3.5 SIEM

Security Information and Event Management の略で，ネットワーク機器やサーバ機器といった，様々なデバイスのログ情報やトラフィック情報を収集し，相関分析を行うことで攻撃を検知する．全体的な傾向を見るのに適している．統計値を使用するために誤差の発生が避けられないことと，デバイスの数が増えると処理する情報量が膨大になり，設備の能力が不足し実時間での処理ができなくなる，また，運用が困難となるなどの課題もある．Splunk[18]，Mcafee SIEM[19]といった商用製品が存在する．

2.3.6 次世代 FW

ファイアウォールに sandbox などの機能を統合したものや．IPS や AntiVirus の機能を統合したものなど，様々なものが存在するが，まだ明確な定義はない．例えば sandbox 型は通信をインタラプトし，FW 上の仮想 OS 上で実行ファイルなどを動作させることで，不正プログラムかどうかを判断する．実際のプログラム動作の結果を確認することで，通信を拒否したり通過させたりすることができる．ただし，標的型攻撃で用いられる不正プログラムで環境依存性があるものは，仮想マシン上の OS やライブラリで動作するとは限らず，

必ず検出できるとは限らない．また，不正プログラムとして検出した場合，それが本当に不正なプログラムかどうかを判断するために高度な解析能力が必要となるなど，実際の運用には多くのコスト（費用，人的）が必要となる．商用として，FireEye[20]，PaloaltoNetwork[21]といった製品が存在する．

2.4 攻撃が成功する要因

このように様々な防御方法があるにも関わらず，標的型攻撃の防御は困難な状況が続いている．本節では，標的型攻撃が成功する要因について，さらなる考察を行う．ネットワークシステム対策の視点から，第 1 章で述べた標的型攻撃の攻撃方法の特徴，また，これまでのシステムの仕組みやその防御方法を踏まえ，攻撃成功要因を以下 3 点にまとめる．

1）内部ネットワークのアクセス制御設計がされていない

ファイアウォールを使って境界防御を行い，アンチウイルスによって不正プログラムから PC を保護できるという前提でネットワークシステムがつくられているため，セキュリティ境界を突破して内部ネットワークに侵入してくる標的型攻撃の対策はそもそも考慮されていない．インターネットとの接続点にファイアウォールを置き，内部ネットワークで不要な機能や通信を制限しないという構造は，いったん侵入を許すと攻撃を止めることも検出することも困難な構造となっている（図 2.3）．

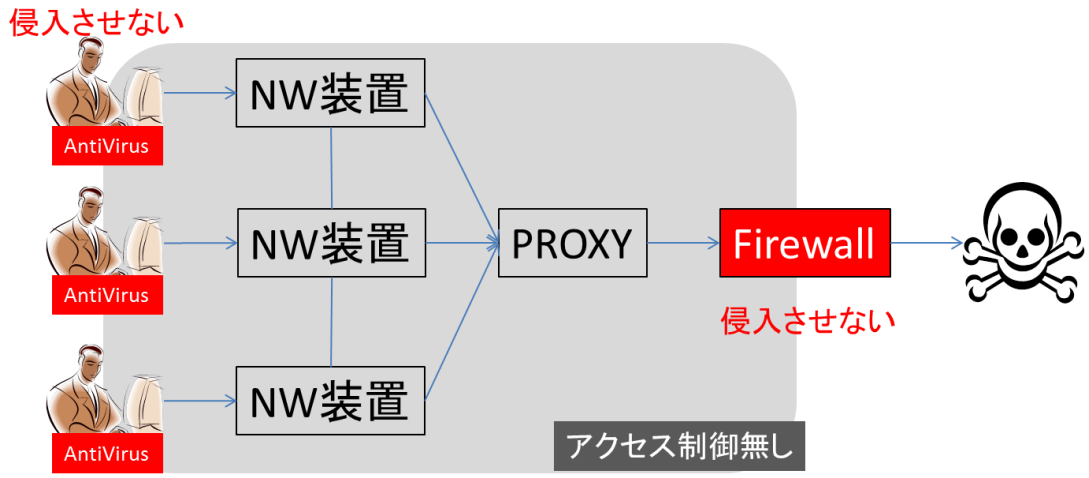

図 2.3 従来のセキュリティ設計

設計者は要件として明示的に示された機能を動作させるために，適切と考えるネットワーク構成を選択し，様々な既製品の OS やソフトウェアを使用することでネットワークシステムのアクセス制御設計を行っている．だが，次々に

新たな機能や製品がリリースされる中，非機能要件を元にそれら既製品が持つ全機能を把握してアクセス制御を設計，構築することは実質的に困難であり，必然的に設計者がよく知る機器設定やネットワーク構成を利用することとなる．その結果，機能要件は満たしていても，必要のない機能や必要のない通信路が存在するかどうかは考慮されなくなる．それらがアクセス制御設計の漏れとして現れる．つまり，機能要件だけを実現する設計ではなく，機能要件は最低限として設計されているが，それ以外の不要な機能も含まれるという設計が行われているのが問題であるといえる．

加えて，TCP/IP で構成されるネットワークは通信レイヤによって実装される機能が異なるため，明示的に機能要件として書かれていなくても，通信を行う上で必要なプロトコルに関しての細かなシステム要件やプロトコルの依存関係が存在する．各レイヤでのアクセス制限や機能の依存関係は，扱うサービスの種類や機器台数が多くなればなるほど加速度的に複雑化し，限られた予算や時間の中で設計を行うには限界がある．設計から実装が行われる際にも，詳細なアクセス制御が設計されていない部分は，使用する製品やソフトウェアのデフォルト設定に依存するため，設計者が意図しない機能が動作することがある．また，現状では上記のとおりアクセス制御設計情報そのものが曖昧さを含むため，実装も曖昧さを含んだものとならざるを得ない．このようなアクセス制御設計方法のため，内部ネットワークでの細かなアクセス制御や，攻撃影響範囲の推定は困難となっている．境界防御を使用して内部ネットワークのアクセス制御を考慮しないという設計は，コストとスケジュールをバランスさせる設計手法としてある意味合理的と言えるが，それゆえの問題点を多く残している．

２）変化する攻撃手法を考慮してシステム設計が行われていない

標的型攻撃は段階的に攻撃手法が変化していくため，偶然にも攻撃が検出できたとしても，検出した時にはすでに攻撃は先の段階へ進んでいる可能性があるため，検出即防御が可能とは限らない．現在のセキュリティデバイスは検出地点で防御を行う，ピンポイントの防御手段であるため，検出ポイントと防御ポイントが異なるような防御を行うことは困難である．このような対策を行うためには，ネットワークシステム全体の構成を把握し，攻撃がどのように行われ，どう進行していくかを網羅的に考慮して，内部ネットワークで適切な防御ポイントとアクセス制御ポイントを設計，実装する必要がある．

３）正規の通信を偽装した攻撃通信を迅速に検知できない

標的型攻撃は持続的に攻撃を行うために，攻撃を行っていることを検出されにくくするよう工夫がされている．侵入をセキュリティ境界で防御するという

発想で作られたネットワークシステムは，外部ネットワークから内部ネットワークへの通信を制限するが，内部ネットワークから外部ネットワークへの通信は「侵入」ではないために制限されない．標的型攻撃はそれを利用し，侵入した端末を使って内部ネットワークから外部ネットワークへと通信を行い，内部ネットワークからの利用に見せかけることで攻撃の検知を困難とし，長期間にわたって内部ネットワークに存在し続ける．

これらの要因によって，標的型攻撃は検出が困難かつ成功する確率が高い攻撃となっていると考えられる．

2.5 既存手法による対策の検討

では，既存の防御手法が使用するに値しないかといえば，そうではない．既存の防御方法で防御可能なものはそれを使用することが妥当である．そこで，既存の防御手法を工夫して適用することによって，標的型攻撃の検出や防御を可能とすることができないか，検討を行った．

まず，一つの対策としては発生する通信すべての通信内容を確認するという方法が考えられる．しかし，そのような方法は，次のとおり様々な観点から現実性が低い．

1）通信の秘密を侵害する可能性が存在する
2）通信量が莫大なため，現実的な時間で処理が難しい
3）莫大なデータ量を処理するため，高性能な解析装置が必要となる
4）通信経路が分散されている場合，経路ごとに検出装置が必要となる
5）事後のフォレンジックを行うためには通信内容を保存しておく必要があり，高億大容量のストレージが必要となる
6）必要な通信が明確に定義されていないため，何が不要な通信かの確認が困難である

次に，内部ネットワークでアクセス制御を行うために中継装置をすべてファイアウォールに置き換えるといった対策も考えられる．しかしこれも次の通り様々な観点から現実性が低い．

1）ルータやスイッチなど，大量の既存通信機器を入れ替えるために高コストとなる
2）ファイアウォールでルータを代替するとインターフェース数が多くなりすぎ，アクセス制御ルールが複雑になり設計が困難となる

３）アクセス制御ルールが機器個別設定であり，全体で連携したルール設計が困難となる

４）個別設定が前提となる機器を統合して管理運用するためには，運用体制の確立やスキルを持った運用エンジニアの確保が必須となる

５） 境界防御内の必要な通信が明確に定義されていないため，止めてよい通信が何かわからない

これらから，既存の防御手法を活用することのみで標的型攻撃を防御するということはやはり困難であるということが言える．

再三述べたように，既存の防御手法では攻撃が行われていることそのものに気が付かない，もしくは気が付けないことが多く，防御も困難となっている．外部からの指摘等で攻撃が行われていることが判明した時には，すでに攻撃が終了しているということも，被害が広がる一つの要因となっている．既存の対策手法であるアンチウイルスによる防御やファイアウォールによる境界防御が困難であるという前提で，可能な限り素早く，内部ネットワークにおいて攻撃の兆候に気が付き，プロアクティブに対応することが重要である．

2.6 従来対策のまとめ

インターネットを取り巻く環境は大きく変化しており，生活を支える情報インフラとして利用されるまでになっている．利用者や機能，装置は莫大な量となり，環境は大きく変化しているにもかかわらず，現在利用されているネットワークシステムのセキュリティ設計は変わらず境界防御とアンチウイルスの配置が主流である．内部ネットワークに侵入されるような新たな攻撃は想定されていない．また，各種のセキュリティ対策装置もバイナリパターンに対応した検出を行い，検出地点で防御を行うという発想で作られている．標的型攻撃はそのような設計手法や実装の弱点を巧みに利用しているため，既存の防御手法では対処が困難となっている．たとえ侵入されたとしても，攻撃の最終段階に到達しない，ネットワークシステム内でも不正な通信を識別可能なネットワークシステムの設計，できるだけ早い段階での攻撃の予兆の検出が必要となっている．

2.7 新たな対策の方向性

これまでに述べた通り，標的型攻撃は既存のセキュリティ対策の限界やネットワークシステムで細かなアクセス制御設計が為されていないことを巧みに利用している．ネットワークシステム設計上から見た根本的な対策としては，内部ネットワークに侵入され，段階的に攻撃が行われることを想定した設計を行

うこと，また，要件上必要とされる通信を偽装している遠隔操作通信をより迅速に検出することが重要となる（図 2.4）．

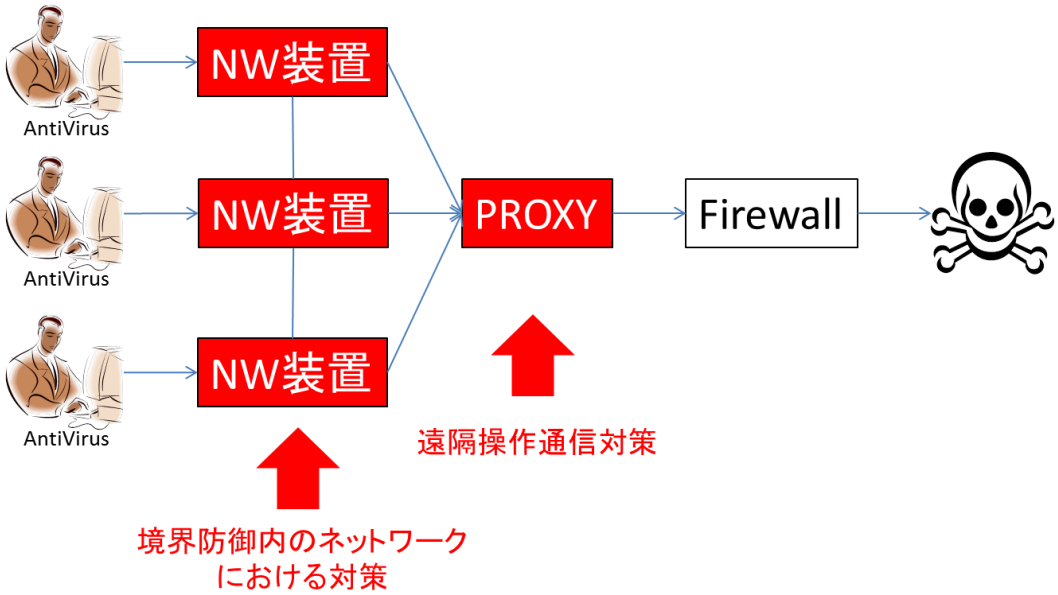

図 2.4 対策の方向性

これらは設計手法，実装手法それぞれで考慮すべき内容であり，従来から行われているセキュリティ対策に加えて，新たな対策を検討する必要があることを示している．そこで，これまでの議論をもとに，新たな対策の方向性を３点にまとめる．

１）内部ネットワークに侵入されたことを前提としたアクセス制御設計

既存のアクセス制御設計手法は，特定機能に絞ったネットワークや物理的なネットワークに対しての設計手法である．先に説明したように，インターネットで使用されるプロトコルはマルチレイヤで複数の機能を実現しているが，マルチレイヤのネットワークシステムにおけるアクセス制御設計手法として次章で詳細を述べるが，確立された手法が存在しない．

よって，内部ネットワークに侵入されたことを考慮し，厳密に内部ネットワークのアクセス制御設計を行うためには，マルチレイヤ・マルチファンクションのネットワーク上でアクセス制御を表現できるネットワーク設計手法が，標的型攻撃への新たな対策として必要と考えられる．

２）段階的に攻撃が進行することを考慮したシステム設計

検出地点と防御地点が同一ではない可能性があること想定して，段階的に攻

撃が進行することを前提として対策を設計する必要があるが，攻撃手法が動的に変化するため，設計者が手作業で網羅的に攻撃過程を検証することは困難である．システム規模の拡大による機能や機器の種類・数量が増えていることも，検証を困難とする原因となっている．設計者の能力に依存して標的型攻撃を考慮した内部ネットワークのセキュリティ設計を行うことは，その多様性から困難となりつつある．

よって，設計者の能力に依存しない設計手法として，設計データを利用した網羅的な攻撃活動のシミュレーション手法が，標的型攻撃への新たな対策として必要と考えられる．

３）偽装された遠隔操作通信の迅速な兆候検出

システムの要件上必要な通信の中から，可能な限り迅速かつ精度が高く，標的型攻撃の兆候としての遠隔操作通信を検出することが重要であるが，現状の対策には限界がある．現状の対策としてアンチウイルスプログラムの使用が提案されているが，標的型攻撃は不正プログラムの流通量が極端に少ないため，アンチウイルスのパターンファイル生成が難しく，その場合検出が困難である．また，サンドボックスを利用するタイプの次世代ファイアウォールが提案されており，これらは実際のプログラムを動作させて標的型攻撃かどうかを判断するため，リアルタイムに近い状態で検出が可能と考えられる．しかし標的型攻撃に使用される不正プログラムは標的に合わせた環境にチューニングされており，OS のバージョンやアプリケーション構成が実際の環境と異なるサンドボックス上で不正プログラムが動作するとは限らないため，確定的に検知できるわけではない．しかも，誤検知かどうかを判断するには不正と判定されたプログラムのバイナリ解析を行わなければならず，運用者の負担が極端に大きいなど運用上の問題もある．

よって，一般に利用される Web アクセスを偽装した標的型攻撃の通信を特定するため，検出精度およびリアルタイム性が高く，運用しやすい手法が，標的型攻撃への新たな対策として必要と考えられる．

2.8 本研究の目的

以上，標的型攻撃の技術的な対策は現状の手法で決め手となるものはないが，これまでに述べた対策の方向性に沿って課題を解決し，システムのセキュリティ設計および実装手法を見直すことで，攻撃への耐性を高めることが防御策となりうると考えられる．

そこで，本研究では 2.7 節の内容に従い，新たな標的型攻撃対策の基礎技術を確立することを目的とすることとする．まず１）および２）に対してである

が，設計，実装といったシステムのライフサイクル上で作りこまれるべき標的型攻撃への対策として，ネットワークシステム内の段階的な探索行為を検出可能とするため，各レイヤにおける機能間の通信可否の明確化ができる設計手法とシミュレーション手法の研究を行うこととする．具体的には攻撃通信の検知を容易にし，攻撃を進行させないためのマルチレイヤで構成されたネットワークシステムのセキュリティ設計手法の研究，および設計情報を用いた標的型攻撃のシミュレーションに関する研究を行う．さらに，３）に対して攻撃の初期段階における攻撃予兆の早期検出技術，具体的には通常の通信を偽装する遠隔操作通信の検出手法に関する研究を行うこととする．（図 2.5）．

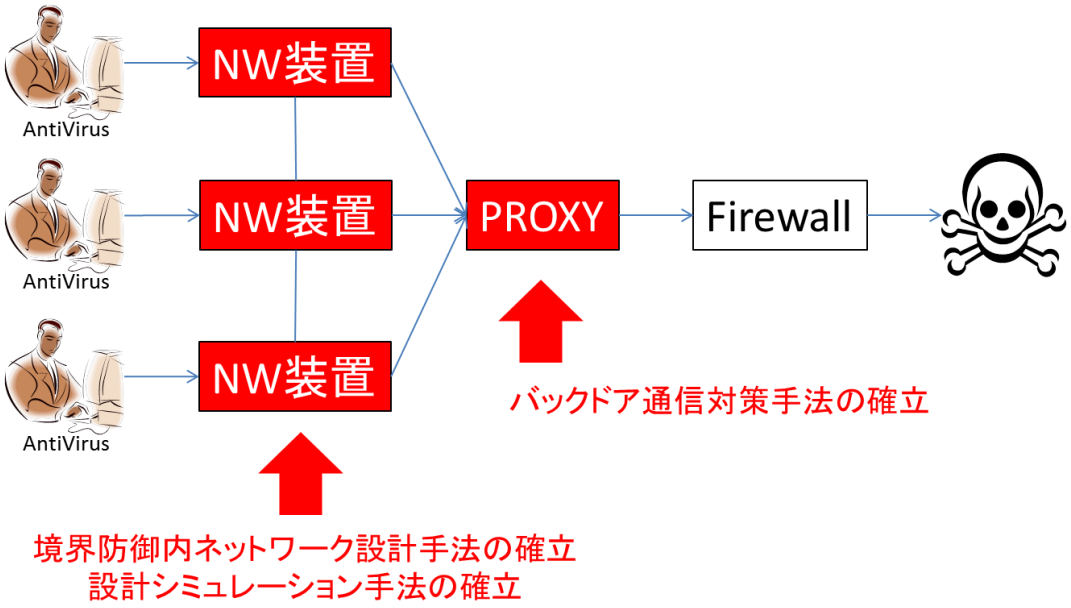

図 2.5 本研究の目的

最後に，表 2.1 に従来対策と本論文の提案手法との対比を示し，その位置づけを明確にする．

	検出	対策
設計	ネットワークシステム内におけるアクセス制御設計の明確化により不要な通信の検知を容易にし，攻撃を進行させない設計手法	ネットワークシステム上へセキュリティ対策装置（FW/AV 等）の配置設計，運用設計

構築 実装	標的型攻撃の初期段階における，攻撃予兆の早期検出技術の研究とその実装	セキュリティ対策装置（FW/AV 等）の設定，インストール
運用	―	設計で定義された手順に従って装置の運用および検知

表 2.1 従来の対策と本論文の提案手法との対比

2.9 本論文の構成

本論文は以下の構成となる．第 1 章ではまず，標的型攻撃が発生する背景として，インターネットの利用目的の変遷，ネットワークシステムの変遷，インターネット上で発生する攻撃の変遷，標的型攻撃がどのような攻撃であるか，などについて概説した．第 2 章では，本論文で想定するネットワークシステムが具体的にどのようなものを指し示すかの前提について解説し，さらにネットワークシステムに対する主な攻撃手法，および実際の対策手法について深堀を行った．加えて，なぜ防御が困難であるかの原因を考察し，その対策の方向性と，対策を実現するための研究内容について述べた．以下，第 3 章では本論文の最も基本となる，ネットワークシステムの新たなセキュリティ設計手法として利用するモデルを提案する．提案ネットワークモデルを使用し，攻撃の影響範囲を定量的に評価可能となることを示す．次に第 4 章では，標的型攻撃が段階的に異なる攻撃手法を使用することによって，ネットワークシステム上でどう攻撃が行われるかを設計段階でシミュレーションするための，プロトタイプ実装を作成する．ネットワークシステムと攻撃手法をドメイン記述言語として記述し，攻撃活動をシミュレーションすることにより，計算による標的型攻撃対策設計の定量的検証が可能となることを示す．さらに第 5 章で，一般ユーザの利用を偽装する通信が行われた場合の検出手法として，通信経路上の情報挿入による偽装通信の検出手法を提案する．標的型攻撃で使用される不正プログラムと一般ユーザの利用における通信上の相違点を説明し，プロキシサーバで意図的にダミーデータとしての HTTP Cookie を通信に混入させることで，一般ユーザ利用と攻撃を識別するという手法を使い，実際に動作環境を構築して検証を行う．その結果，ネットワークシステム侵入後の標的型攻撃通信がリアルタイムに検出されることを示す．最後に第 6 章において本論文全体のまとめと，残された課題，および，将来的な標的型攻撃対策のあるべき形について言及する．

参考文献

［１］情報処理推進機構：コンピュータウイルス・不正アクセス届出状況および相談受付状況[2013 年年間]，情報処理推進機構（オンライン），入手先〈http://www.ipa.go.jp/security/txt/2014/2013outline.html〉（参照 2014-10-29）
［２］JPCERT コーディネーションセンター：Web サイト改ざん及びいわゆる Gumblar ウイルス感染拡大に関する注意喚起，JPCERT コーディネーションセンター（オンライン），入手先〈https://www.jpcert.or.jp/at/2010/at100001.txt〉（参照 2014-10-29）
［３］Cyber Froce Center: ウェブサイト改ざん事案の再多発に係る注意喚起について，警察庁（オンライン），入手先〈https://www.npa.go.jp/cyberpolice/detect/pdf/20130930.pdf〉（参照 2014-10-29）
［４］情報処理推進機構：サービス妨害攻撃の対策等調査，情報処理推進機構（オンライン），入手先〈http://www.ipa.go.jp/files/000014123.pdf〉（参照 2014-10-29）
［５］Steve Linford: An arrest in response to March DDoS attacks on Spamhaus, The Spamhaus Project Ltd.(online), available from〈http://www.spamhaus.org/news/article/698/an-arrest-in-response-to-march-ddos-attacks-on-spamhaus〉(accessed 2014-10-29)
［６］IPTABLES online manual，入手先〈http://linuxjm.sourceforge.jp/html/iptables/man8/iptables.8.html〉(参照 2014-10-29)
［７］Home of the pfSense Project, Electric Sheep Fencing LLC(online), available from〈https://www.pfsense.org/〉(accessed 2014-12-23)
［８］Firewall-1, Check Point Software Technologies Ltd.（オンライン），入手先 〈http://www.checkpoint.co.jp/products/firewall-1/〉(参照 2014-10-29)
［９］SSG シリーズ，Juniper Networks Inc.（オンライン），入手先〈http://www.juniper.net/jp/jp/products-services/security/ssg-series/〉（参照 2014-10-29）
［１０］SNORT, Cisco Systems, Inc.(online), available from〈https://www.snort.org/〉(accessed 2014-10-29)
［１１］IBM Security Network Intrusion Prevention System，日本 IBM（オンライン），入手先〈http://www-03.ibm.com/software/products/ja/network-ips〉（参照 2014-10-29）
［１２］Microsoft Security Essentials で使用中の PC を保護する，日本マイクロソフト（オンライン），入手先〈http://www.microsoft.com/ja-jp/security/pc-security/mse.aspx〉（参照 2014-10-29）
［１３］トレンドマイクロ株式会社，入手先〈 http://www.trendmicro.co.jp/jp/index.html〉（参照 2014-10-29）
［１４］株式会社シマンテック，入手先〈http://www.symantec.com/ja/jp/〉

（参照 2014-10-29）
［１５］株式会社カスペルスキー，入手先 <http://www.kaspersky.co.jp/> （参照 2014-10-29）
［１６］OCN DDoS 対策サービス，エヌ・ティ・ティ・コミュニケーションズ株式会社（オンライン），入手先 <http://www.ocn.ne.jp/business/security/ddos/> （参照 2014-10-29）
［１７］IIJ DDoS プロテクションサービス，株式会社インターネットイニシアティブ（オンライン），入手先 <http://www.iij.ad.jp/biz/ddos/> （参照 2014-10-29）
［１８］splunk，Splunk Inc.(online)，available from <http://www.splunk.com/> (accessed 2014-10-29)
［１９］McAfee Security Information and Event Management (SIEM)，McAfee Inc.（オンライン），入手先 <http://www.mcafee.com/japan/products/siem/> （参照 2014-10-29）
［２０］FireEye: FireEye，FireEye Inc. (online)，available from <http://www.fireeye.com/> (accessed 2014-10-29).
［２１］PaloAltoNetworks: Next-Generation Firewalls，Palo Alto Networks Inc. (online)，available from <https://www.paloaltonetworks.com/> (accessed 2014-12-23).

第3章 セキュリティ設計のためのマルチレイヤネットワークモデルの提案

3.1 背景

第 2 章で述べたように，ネットワークシステムの安全性に関する設計は境界防御とアンチウイルスによるものが中心で，設計者の経験に依存して行われており定性的である．方法論や理論などの再現性をもった定量的な評価はほとんど行われていない．定性的な設計，評価，検証による曖昧さを排除するためには，定量的に評価可能なシステム設計手法が必要である．

ネットワークの安全性に関する研究は多くされてきた[1][2][3]．またネットワークの最適設計に関する研究も多くされている．しかしこれらの研究は，バックボーンネットワークといった Wide Area Network (WAN) に対する研究である．さらに多くの既存研究は単一種類の機器によるネットワークや単一レイヤでのネットワーク構成を前提としているため，サーバやルータ，スイッチ，ファイアウォール，ロードバランサなど，構成機器がそれぞれ異なるセキュリティ機能を提供しその結合によりサービスを提供するネットワークシステムとは大きく前提条件が異なる．そのため，標的型攻撃のように，ネットワークシステム上で多種多様な通信を行うような攻撃へのシステム設計対策として，従来研究の成果を適用することは困難である．

標的型攻撃対策における，ネットワークシステムの安全性に関する最適設計を検討する場合には，まず安全性の定義と，何が安全かを示す定量尺度が必要である．定量尺度の存在により，最適な安全性設計の議論が可能になる．そしてその尺度も，評価者によって測定値が異なることのないよう，同一のシステムと同一の条件下では同一の測定値が得られる必要がある．

システム測定値の同一条件下での一意性を保証するためには，測定に用いられるシステムは，曖昧さを排除した表現が為される必要がある．特にネットワークシステムの場合，複数の異なる機能を持つ機器群によりシステムが構成されるため，曖昧さを排除するためにはそれら複数機器の種類が特性を失うことなく表されることが必要になる．また，TCP/UDP のポート番号に従ってパケットのフィルタリングを行うファイアウォールや IP アドレスに従ってルーティングを行うルータなど，システムにおける各機器の機能が異なるレイヤで実現されていることから，各機能を表現するには複数のレイヤ情報を包含した表現が求められる．また上位レイヤの機能を実現するには関連する下位レイヤの機能も必要となり，各レイヤにまたがる機能の依存も不足なく表現される必要がある．

そこで本章では標的型攻撃に対する設計上の対策として，複数機器・複数機

能の相互接続により構成されるネットワークシステムに対し，それらの機器の機能特徴を失うことなく論理的にアクセス制御情報を表現可能なマルチレイヤネットワークモデルを提案する．提案ネットワークモデルはこれまでのグラフ理論によるグラフ表現にレイヤ構造を取り入れて拡張したモデルであるが，各ノード間にはレイヤ構造による依存関係があるほか，ノードを結ぶリンクは通信路を意味するだけでなくシステム内の中継機能や依存関係など複数の意味を持つ．レイヤ構造の適用と依存関係の反映により，提案ネットワークモデルはこれまでのネットワークモデルでは表現が困難であった複数機器・複数機能の柔軟な表現を可能にした．また提案ネットワークモデルは論理ネットワークだけではなく物理ネットワークを含むことで，物理的な構成によるリソース制限の論理的な構成への影響を検討可能にした．さらに，提案ネットワークモデルはサーバなどの仮想化と実体の依存関係も表現可能であり，クラウドなどの仮想化環境にも対応している．

定式化による集合の操作により各レイヤのネットワークを部分的に抽出することで，安全性に関する設計だけでなく，単一レイヤ・単一種類ノードで実現される既存のネットワーク設計理論における種々の手法を適用することも可能である．しかし適用可能なのはレイヤごとに抽出されたネットワークのみであり，マルチレイヤ全体への適用は難しい．その理由に，最適を示す尺度がマルチレイヤのモデルに適していないことがある．例えば，これまでのネットワーク設計理論で研究されてきたフロー問題では尺度としてフロー流量が用いられ，システム全体の尺度としてはフローの総量が利用される．しかし，マルチレイヤ特性をもつ提案ネットワークモデルでは各レイヤのフロー流量は他のレイヤと強い依存関係を持つことからレイヤ種類を考慮しないフロー総量は依存による重複が起こり，システムのフロー流量を判断するには不適切である．これら尺度についてもレイヤ構造や依存関係などマルチレイヤの特徴を考慮したものにならなければ，その尺度を用いた最適設計手法を検討することはできない．

本章ではモデルの提案に加え，提案ネットワークモデルを利用した安全性の定量評価尺度も併せて提案する．定量評価尺度はネットワークシステム内で，初期潜入段階において脆弱性を持ったクライアントが侵入された後に，攻撃が拡散する規模を測る指標として，脆弱性影響度の提案を行うこととした．

3.2 節ではネットワーク設計についてこれまで行われてきた従来研究を紹介し，3.3 節で従来研究では実現されていなかったマルチレイヤネットワークモデルを提案する．3.4 節では提案ネットワークモデルを利用した評価尺度の検討と，実際の機器との対応について述べ提案ネットワークモデルの有効性を示し，3.5 節でまとめる．

3.2 関連研究

ネットワークの最適設計は古くから行われている研究分野であり，近年においても，Belotti らが複雑なノードコストを持つネットワークの設計問題についての解法を提案し[4]，Chekuri らはフローギャップが単一であるケースでの頑健なネットワーク設計を行い[5]，また El-Alfy が MPLS ネットワークにおける最小コストトポロジーを遺伝的アルゴリズムで求めるなど[6]，多くの研究が行われている．

しかしこれらの研究が対象としているネットワークはノードの種別が単一であり，多数の機能を持った機器が相互作用するネットワークの設計を対象にしているものではない．また，単一のノード種別ではないものであってもレイヤ構造を持つものではないものが多い[7][8]．

一方，レイヤ構造を持ったモデルを検討している研究もある．Belotti らは MPLS ネットワークの設計において，論理的なノードによるネットワークと物理的なノードによるネットワークの 2 階層を考慮した設計手法を提案している[9]．また Dijkstra らは ITU-T G.805 をもとにした多層構造を持つネットワークのモデルを提案している[10]．しかし，Dijkstra らの提案は Belotti らと同様に MPLS のモデル化であり，MPLS 機器同士のネットワークはレイヤ構造を持つが単一のノードで構成されているものであることから，複数機器の違いを包含可能なモデルとは言えない．これらモデルは ISP などのネットワーク事業者が持つバックボーンネットワークを対象にした大規模なネットワークモデルであり，本研究で対象としているサービス提供用ネットワークシステムやクライアントネットワークシステムにあるような，サーバを含む複数の機器を表現し，設計へ応用可能にしているものではない．

一方，Salvador らは様々な通信が行われるローカルエリアネットワークのモデル化を行っているが，ネットワークトポロジはモデルに含まれずネットワーク全体の機能を抽象化したものとなっている[11]．

また，従来のノード費用やフロー費用を尺度としたネットワーク設計だけでなく，他の尺度を用いた最適設計の研究も行われている．Habib はネットワーク再設計でのコスト最適化手法を提案し[12]，そこでは機器を複数扱うことやポート数やスループット性能，価格等の属性を適用するなど，複数尺度での最適化を提案している．

従来はレイヤ構造を持ったモデル化や，ノード種別を複数持つネットワークのモデル化，あるいはフローやノード費用以外の尺度を用いたネットワーク設計手法など，個々の関連研究は本研究が対象とするネットワークシステムの安全性設計に関連するが，レイヤ構造を持ちノード種別が複数存在するネットワークのモデルや設計手法は存在していない．さらに個々の機器が持つ脆弱性に

対するネットワークシステムの安全性設計へのアプローチはなされていない．

3.3 マルチレイヤネットワークモデル

ネットワークシステムとは第 2 章で示したように，複数の機器がローカルエリアネットワーク（LAN）技術によって接続されネットワークを構成し，ネットワーク全体で 1 つ以上のサービスを提供しているシステムをいう．

本節では，ネットワークシステムを構成する各機器の特徴を失わない新たなネットワークモデルを提案する．提案ネットワークモデルは，これまでのマルチレイヤモデルでは表現が難しかった依存関係の明確化や，複数種類の機器を同一モデル内に表現可能とするものである．提案ネットワークモデルはグラフ理論でのグラフ $G=(V, E)$ を拡張したものであり，ノードとリンクの集合と 4 つの写像で表現される．しかし，既存モデルとは異なり，1 つの機器は 1 つのノードでは表されず，機能の要素としてノードが各レイヤに存在し，それらノードとリンクの集合（部分グラフ）により 1 つの機器（モジュール）を表現する（図 3.1）．モデルの定義はレイヤ定義，ノード定義（属性定義），リンク定義（属性定義），モジュール定義，ネットワーク定義で行われる．次にそれぞれの定義について説明を行う．

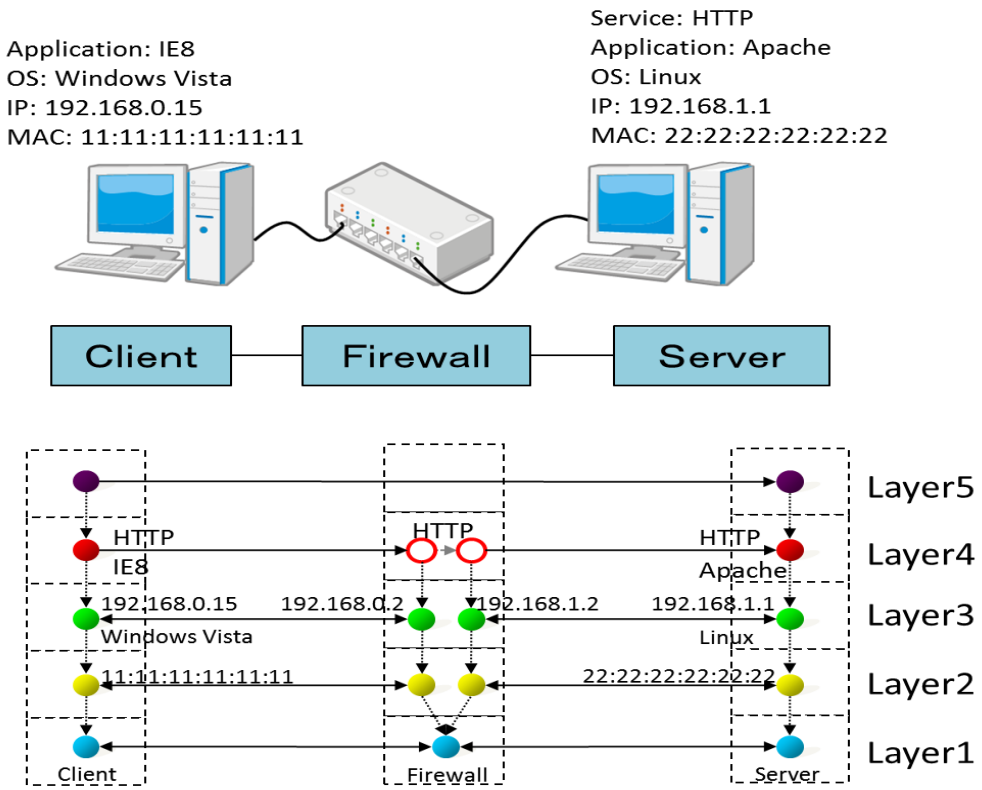

図 3.1 提案ネットワークモデルによる表現例

3.3.1 モデルの定義

定義 3.1[レイヤ]

レイヤは 5 つからなる．レイヤ 1（L1）は物理的接続の層であり，レイヤ 2（L2）は Ethernet ネットワークの層，レイヤ 3（L3）は IP ネットワーク層，レイヤ 4（L4）は TCP/UDP ネットワーク層，そしてレイヤ 5（L5）は抽象化したサービスの層である（表 3.1）．

Layer 5	抽象化サービス（WWW，電子メール，etc.）
Layer 4	TCP/UDP ネットワーク（port 80，25，etc.）
Layer 3	IP ネットワーク
Layer 2	Ethernet ネットワーク
Layer 1	物理接続

表 3.1 レイヤ定義

通信におけるレイヤの定義は ISO 7498 で定義されている OSI 参照モデルによる 7 階層や，RFC 1122 で定義されている TCP/IP による 4 階層が代表的であるが，本論文で定義するレイヤは TCP/IP をもとに，ネットワークシステム設計で必要とされる物理的な接続のレイヤを最下位レイヤ，要件として定義されるサービスを最上位レイヤとして，5 階層としたものである（図 3.1）．

定義 3.2[ノード]

通信の終端あるいは中継点となる要素をノードと呼ぶ．ノードは終端ノードと中継ノードの 2 種類が存在する．

終端ノードは通信の始点または終点となるノードであり，中継ノードは通信の始点あるいは終点ではないが通信を行うにあたり始点のアイデンティティ情報と終点のアイデンティティ情報からデータ配送可否の判断や配送する通信路の決定を行うノードである．

ノードは 3 つの属性を持つ．1 つはレイヤ情報，もう 1 つは終端ノードか中継ノードの種別情報，最後にアイデンティティ情報である．アイデンティティ情報はシステム内で一意に識別されるための情報で，IP アドレスや MAC アドレス，ポート番号などが適用される．

レイヤ 2 以上に属するノードは，必ず 1 階層下のノードと依存関係リンクにより接続されていなければならない．依存関係リンクに関しては後述する．

ネットワーク上の各ノードは v_i，その集合は V で表される．また 2 つの属性

情報を持つため，レイヤ情報を示す写像 l_V: $V \rightarrow L_V$，ノード種別を示す写像 s_V: $V \rightarrow S_V$ が存在する．ここで $L_V = \{1, 2, 3, 4, 5\}$，$S_V = \{T, R\}$ である．L_V の各要素はレイヤ情報を示すものである．また S_V の要素 T は終端ノード（Terminal）であることを示す情報であり，R は中継ノード（Relay）であることを示す情報である．

定義 3.3[リンク]

ノード間を結ぶ要素をリンクと呼ぶ．リンクは通信路リンクと依存関係リンク，中継リンクの 3 種類よりなる（図 3.2）．

通信路リンクは同一レイヤでの異なるモジュールに属するアクセス可能なノード間を結ぶリンクであり，当該レイヤでの通信路を示すものである．モジュールに関しては後述する．通信路リンクは向きを持ち，通信の方向を示す．レイヤ 2 以上に属する通信路リンクは，当該リンクの始点・終点となるノードの各下位ノード間で到達可能になっていなければ存在できない．

依存関係リンクは，依存関係があるノードを結ぶリンクである．レイヤ間を結ぶことも可能であるがノード間のレイヤ属性値の差は 1 に限る．また依存関係リンクは，依存ノードから被依存ノードへの向きを持つ．

中継リンクはモジュール内で同一レイヤの中継ノード間を結ぶリンクである．中継リンクは向きを持ち，中継の方向を示す．

リンクは 2 つの属性を持つ．1 つはレイヤ情報であり，もう 1 つは通信路リンク，依存関係リンク，中継リンクのリンク種別を示す種別情報である．

ネットワーク上の各リンクは $e_i = (v_a, v_b)$ で表される．ここでは e_i はノード v_a から v_b へのリンクであることを示している．リンクの集合は E で表される．また 2 つの属性情報を持つため，それぞれに写像 l_E: $E \rightarrow L_E$，s_E: $E \rightarrow S_E$ が存在する．ここで $L_E = \{0, 1, 2, 3, 4, 5\}$，$S_E = \{C, D, R\}$ である．L_E の各要素はレイヤ情報を示すものであり，0 はレイヤ情報を持たないことを示すものであり依存関係リンクの属性値として用いられる．S_E の要素 C は通信路（Communication）リンクを示し，D は依存関係（Dependency）リンク，R は中継（Relay）リンクを示すものである．ノード間にリンクが存在しなければ通信は設計上許可されていない．

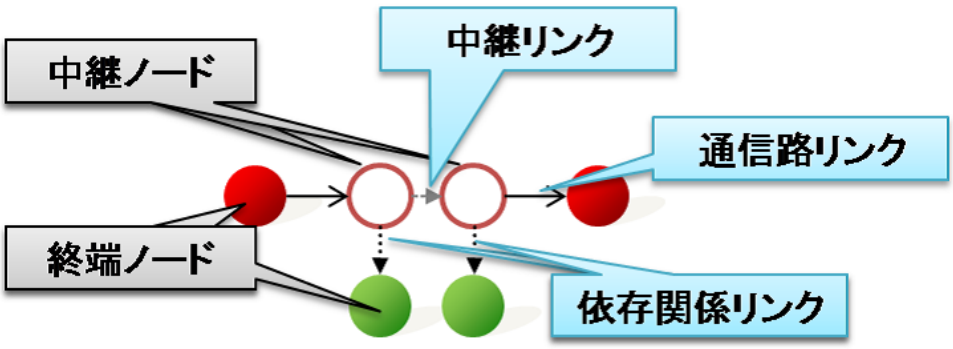

図 3.2 ノードとリンクの種類

定義 3.4[モジュール]

ネットワーク機器の機能を提供するものをモジュールと呼ぶ．モジュールは依存関係リンクと中継リンク，ノードにより構成される．

モジュールはその機能によりサービスモジュール，インターネットモジュール，中継モジュールの 3 つに大別される．中継モジュールはレイヤごとに細分化して表される（図 3.3）．

・サービスモジュール（S）
・インターネットモジュール（I）
・中継モジュール（R）
　– レイヤ n 中継モジュール（LnR）

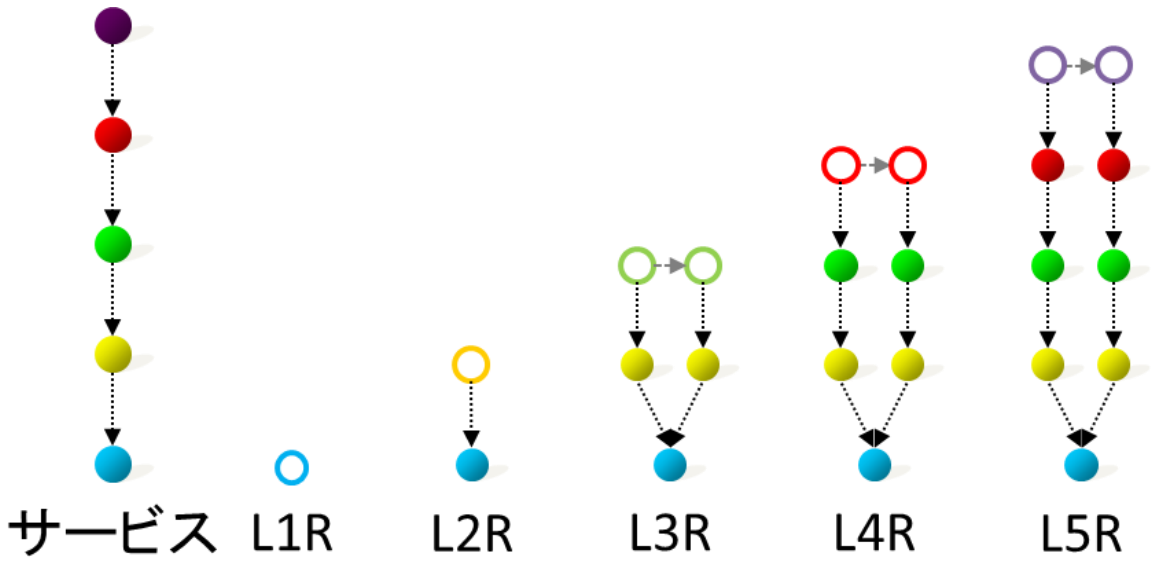

図 3.3 モジュール例

サービスモジュール（S）はサービスを提供するモジュールである．また他のサービスモジュールに対するクライアントになることも可能である．各レイヤのノードはアイデンティティ情報を持つ．

インターネットモジュール（I）は外部ネットワーク（インターネット）を代表するモジュールである．各レイヤのノードは複数のアイデンティティ情報を持つことが可能である．

中継モジュール（R）はあるレイヤでの通信の中継を行うモジュールである．レイヤにより 5 つに細分化される．レイヤ 2 以上のノードはアイデンティティ情報を持たないことがあり，またモジュール内の最上位レイヤに存在するノードはすべて中継ノードとなる．

各モジュールが持つノード数は表 3.2 のようになる．ここで n_i はそのモジュールのレイヤ i でのノード数を示す．また抽象化されたサービスを示すレイヤ 5 のノードは，複数のモジュールより構成されるものであるため，その個数はモジュールごとには定まらない．よって表には記載されていない．

	S	I	R				
			L1R	L2R	L3R	L4R	L5R
L4	$\geq n_3$	1	0	0	0	$\geq n_3$	$\geq n_3$
L3	$\geq n_2$	1	0	0	n_2	n_2	n_2
L2	≥ 1	1	0	≥ 1	≥ 1	≥ 1	≥ 1
L1	1	1	1	1	1	1	1

表 3.2 各モジュールが持つノード数

定義 3.5[ネットワークとシステム]
ネットワークは，ノードとリンクの集合であり，システムは各レイヤのネットワークを結合したものである．ネットワーク G は，以下で定式化される．

$$G = (V,\ E,\ l_V,\ s_V,\ l_E,\ s_E) \tag{1}$$

3.3.2 定式化によるネットワーク操作

定義とともに示した定式化により，ネットワーク G に対して部分グラフとして特定レイヤのネットワークを抽出することや，ノード部分集合の抽出といった操作が可能になる．

レイヤ x に属するノード集合 V_x とリンク集合 E_x はそれぞれ以下のように表すことができる．

$V_x = \{v_i \mid l_V(v_i) = x\}$ (2)
$E_x = \{e_I \mid l_E(e_i) = x\}$ (3)

またこれらから，システム全体のネットワーク G からレイヤ x のネットワーク G_x を抽出することが可能である．

$G_x = (V_x, E_x, l_V, s_V, l_E, s_E)$ (4)

さらに，リンク種別ごとの集合 E_C, E_D, E_R を以下のように抽出することも可能である．

$E_C = \{e_I \mid s_E(e_i) = C\}$ (5)
$E_D = \{e_i \mid s_E(e_i) = D\}$ (6)
$E_R = \{e_I \mid s_E(e_i) = R\}$ (7)

上記の各集合により，ネットワーク G が各レイヤの論理ネットワーク G_i と各ノードの依存関係のみを表現したネットワーク（$V, E_D, l_V, sV, l_E, s_E$）で構成されることがわかる．

$$G = (V, E_D, l_v, s_v, l_E, s_E) \cup \bigcup_i G_i \quad (8)$$

3.3.3 モデルによる利点

既存研究においてもマルチレイヤネットワークのモデルは存在するが，標的型攻撃対策として行うネットワークシステム設計への適用には向いていない．たとえば，マルチレイヤモデルで表現されたシステムから，システム内に存在するファイアウォールのルールを抽出することを考える場合，通信路の始点と終点が明確にされていなければルール抽出は不可能である．既存モデルではノードが区別されておらず，ファイアウォールもサーバも同一レイヤで同一ノードとして表現されるため始点と終点を明確に表現できないが，提案ネットワークモデルでは正確にファイアウォールのルールを抽出可能である．このように，提案ネットワークモデルによる設計情報は現実機器・設定への適用といった運用，設計情報を用いたシミュレーションなどに対応可能である．

近年のクラウドコンピューティング環境で主流になりつつある仮想化にも提案ネットワークモデルは対応している．依存関係リンクを利用することにより，

依存ノードにゲスト PC の L1 ノード，被依存ノードにホスト PC の L1 ノードを用意し，両者を依存関係リンクで結び，さらに仮想ブリッジとして L1 ノードを設け，仮想ブリッジノードからホスト PC の L1 ノードに依存関係リンクを結ぶことで，ホスト・ゲストの依存関係の表現が可能である．そして上位レイヤではそれら依存関係を考慮することなくサービス構成を行うことが可能になる．

さらに，冗長化の面においても仮想 IP アドレスを用いるロードバランサや，仮想ルータにも対応可能である．仮想 IP アドレスを L3 ノードとして設け，それらと実際の提供サービスホストとを依存関係リンクで結ぶことで冗長化されたシステムの表現が可能になる．

提案ネットワークモデルによる柔軟な表現の対応は，一方でモデルの複雑さを招き，利用者にとって可読性の低いものとなる可能性がある．しかし利用者は本質的にモデル自体を閲覧する必要はなく，モデルを用いた各アプリケーションが，その用途に応じて必要な情報のみを提示することで可読性が高められるべきである．

3.3.4 モデルによる表現の例

図 3.4 のネットワーク図で表されるシステムを，提案ネットワークモデルで表現した例を図 3.5 に示す．図 3.4 のネットワーク図は機器の物理的つながりを示したネットワークでしかなく，IP ネットワークなどの論理的な接続やセグメント表現はできていない．標的型攻撃対策設計で必要となる論理的な接続情報やアクセス制御情報は，関連性がなく表記も異なる個別の情報として記述される．一方，提案ネットワークモデルを用いた図 3.5 では，物理的つながりはレイヤ 1 で表され，上位の論理的なネットワークも表現されており，さらにそれらの依存関係も表現されていることがわかる．なお，図 3.5 では，各レイヤは色分けして表現してあるが，外観の煩雑さを除くために各リンク種別とノード種別の差，またリンクの向きは表現していない．

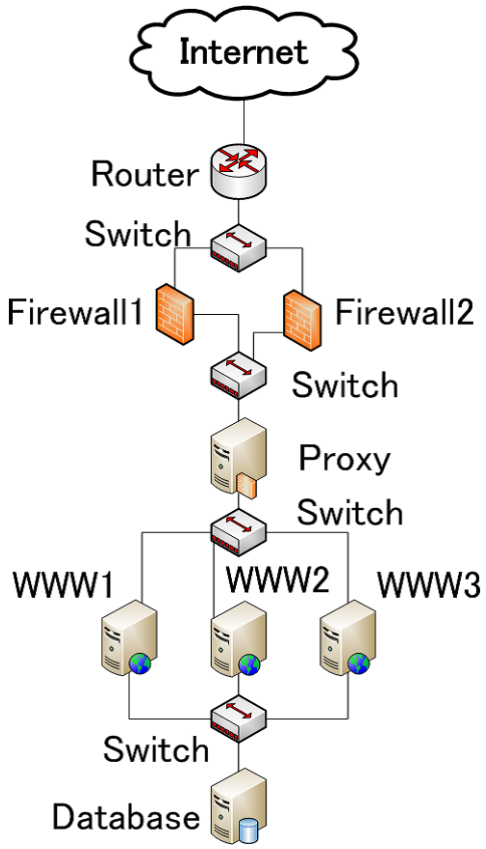

図 3.4 ネットワークシステムのネットワーク図

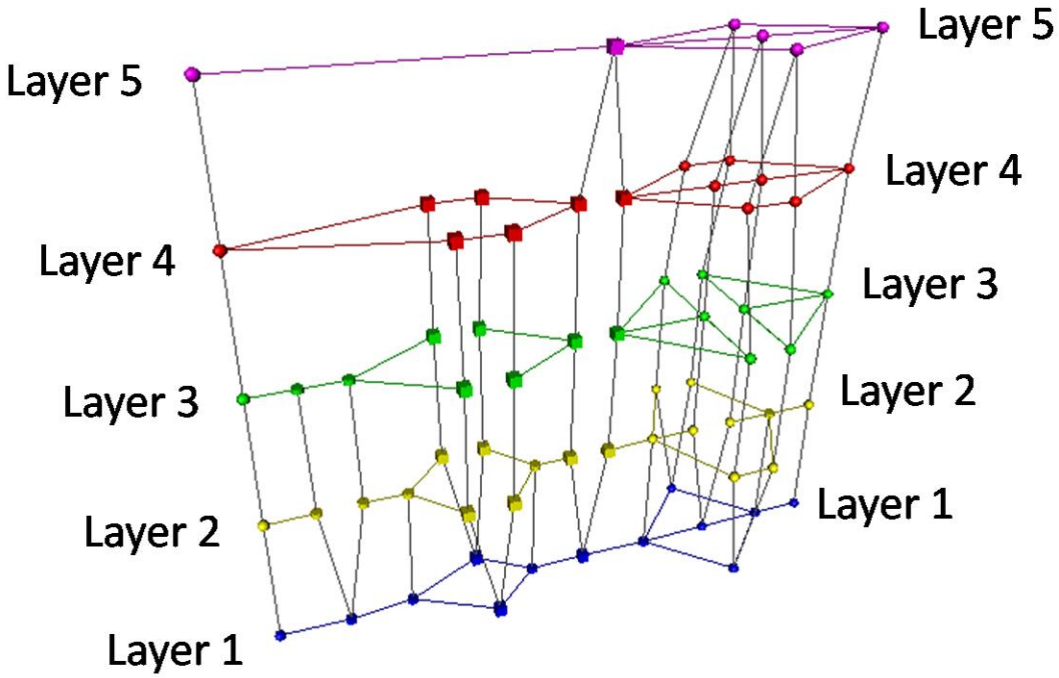

図 3.5 ネットワークシステムのモデルによる表現

3.4 提案ネットワークモデルの応用

3.4.1 最適性と評価尺度

ネットワーク設計の最適性を論じるには，最適を示す尺度が必要である．そしてその尺度を用いて，ネットワークの最適性を求める問題を解くことがネットワーク設計問題である．そこでまず，内部ネットワークでの探索や拡散活動が既存のネットワーク設計理論で評価可能かどうかを検討した．

これまでのネットワーク設計理論では，フロー費用を尺度とした最小費用フロー問題や，利用者均衡フロー問題，システム最適化フロー問題などがあり，予算を尺度とした予算制約を持つネットワーク設計問題（Budget Network Design problem）や固定費用を持つネットワーク設計問題（Fixed charge Network Design problem）などがある[13]．これらはレイヤ構造を持たない既存のネットワークモデルでの問題であり，提案ネットワークモデルを用いた最適設計を論じるには，これら尺度をマルチレイヤに拡張する必要がある．

さらに安全性の視点では，システムの最適性はアクセス制御の状態，各機器の脆弱性がシステムの安全性にどこまで影響を及ぼすか，各機器の障害がシステムにどのように影響するか，といったフロー費用や予算などとは異なる様々な視点で検討される必要がある．

このように最適性を論じるにはその視点が重要になるが，提案ネットワークモデルにおける最適性の検討では下記の 2 点が考慮されなければならない．

・標的型攻撃に対する安全性視点でのネットワーク設計問題の設定
・既存設計問題のマルチレイヤモデルへの拡張

本節では，上記 2 点のうち「標的型攻撃に対する安全性視点でのネットワーク設計問題の設定」に注目し，各ノードが持つ脆弱性がネットワークシステムの安全性にどれほどの影響を及ぼすか（脆弱性影響度）の定量尺度の検討を行うことを目的とする．既存の設計問題では各問題についてその解法に先立つ計算困難性について論じられており，提案ネットワークモデルを初めとしたマルチレイヤモデルへの拡張にあたっては計算困難性から再検討しその解法を議論する必要があるため,今後の課題とする．

3.4.2 脆弱性の影響度

脆弱性の影響度に関しては，CVSS（Common Vulnerability Scoring System）による測定が知られている[14][15]．CVSS は脆弱性単体の危険度を示す基本値（Base Score）や,脆弱性を利用するワームや脆弱性を無効化するパッチの存

在など時間経過により異なる脆弱性の危険度を示す現状値（Temporal Score），そしてシステムやネットワークといった環境全体への脆弱性の危険度を示す環境値（Environmental Score）の３つの尺度からなる．

CVSS の基本値は米国標準技術局（NIST）がもつデータベース NVD（National Vulnerability Database）[16]に含まれるすべての脆弱性情報に付与されていることや,日本でも情報処理推進機構（IPA）による脆弱性対策情報データベース JVN iPedia[17]でも採用されるなど広範に利用されている一方で,現状値や環境値はほとんど利用されていない．

さらに，環境値は自分の環境情報として「影響を受ける対象システムの範囲（TD:Target Distribution)」を入力しなければその値が出力されないが，その入力すべき情報が「なし」「小規模（利用環境の 1-25%)」「中規模（利用環境の 26-75%)」「大規模（利用環境の 76-100%)」と非常に粗い尺度でしか設定ができない．それらの範囲測定についても明確な基準がなく，再現性が高いとは決して言えない．そこで，本モデルを利用することで曖昧さを排除可能な環境値への TD 入力手法を提案する．提案にあたり，ネットワークシステムは物理的に保護されていることとした．そのため，CVSS の基本評価基準内の「AV:攻撃元区分」において物理的なアクセス可能状態からの攻撃を必要とする「ローカル」は考慮しない．

まず脆弱性とモデルのマッピングを行う．脆弱性はモデル上のノードがそれぞれ保持することとし,脆弱性が持つ属性情報と各レイヤのノードを表 3.3 のように対応付けた．

レイヤ	属性
5	アプリケーションデータ
4	アプリケーション名，バージョン
3	OS 名，バージョン
2	-
1	製品名，ベンダ名

表 3.3 属性のマッピング

ネットワークシステムがファイアウォールなどによりアクセス制御が行われている場合，脆弱性が存在したとしても攻撃が該当機器まで到達せず，脆弱性の影響を受けない可能性がある．そのため，単純に脆弱性のあるアプリケーションが内部ネットワークに存在するというだけでは影響は判断せず,システム外部（インターネットモジュール）から到達可能であるノードかつ脆弱性を持

つノードを攻撃対象ノードとした.

そして攻撃対象ノードのシステム内での位置と，脆弱性の性質により影響範囲を決定する．脆弱性の性質は CVSS の基本評価基準（Base Metric）に沿って機密性への影響（C: Confidentiality Impact），完全性への影響（I: Integrity Impact），可用性への影響（A: Availability Impact）を考慮する．

機密性は，サービスが抱える情報の機密性と，サービスを構成している各機器が持つ設定情報やユーザ情報などの機密性とに大別できる．本モデルはサービス上のデータ自体ではなく，サービスを構成する機器とネットワーク構成を示すことから，ここでは後者を考える．完全性も同様にサービス自体の完全性と，サービス提供を行う機器が持つ情報の完全性とに大別でき，ここでは後者を考える．一方，可用性は構成する機器やソフトウェアの可用性が保たれることでサービスの可用性が実現されるため，分離して考えることはしない.

CVSS において各脆弱性の C，I，A 評価はそれぞれ「なし」「部分的」「全面的」の 3 段階で評価される．提案手法では各ノードの依存関係を考慮し，「部分的」である場合は脆弱性が存在するノードの上位に存在するノード（依存関係リンクで結ばれた上位ノード）も影響を受ける．「全面的」である場合は脆弱性が存在するノードが所属するモジュール全体（当該ノードから依存関係リンクのみで到達可能なすべてのノード）が影響を受けるとし，それらノード集合を「初期脆弱ノード群」と呼び，V_{initC}，V_{initI}，V_{initA} で表す．さらに，脆弱性のシステムの影響は初期脆弱ノード群だけにとどまらず周辺にも影響を及ぼすことから，初期脆弱ノード群からリンクで繋がっているノード群（以後，周辺ノードと呼ぶ）を「脆弱性影響ノード群」とし，V_C，V_I，V_A と表す．機密性（C）と完全性（I）は「通信リンクを通じて隣接しているノードが影響を受ける」を周辺ノードとし，可用性（A）については「通信リンクを通じて到達可能なノードすべてが影響を受ける」を周辺ノードとした.

図 3.6 は C と I に「部分的」な脆弱性を持つ場合の初期脆弱ノードと周辺ノード群を示したものである.

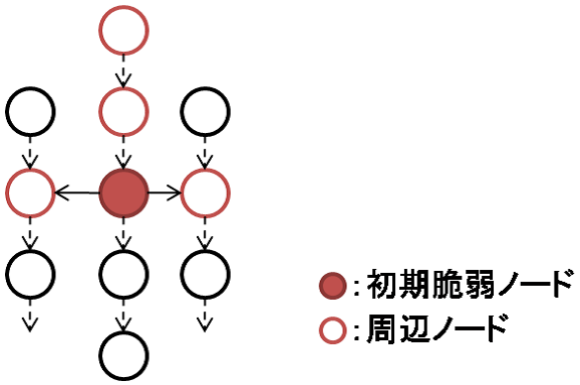

図 3.6 初期脆弱ノードと周辺ノード

V_C，V_I，V_Aは以下のように表される．

$$\mathrm{V}_C = \mathrm{V}_{initC} \bigcup \{v_i | (v_i, v_x) \in E_C\}$$
$$\mathrm{V}_I = \mathrm{V}_{initI} \bigcup \{v_i | (v_i, v_x) \in E_C\} \quad (9)$$
$$\mathrm{V}_A = \mathrm{V}_{initA} \bigcup V_{reachableA}$$

ここで $V_{reachableA}$は V_{initA}から通信リンクのみを利用して到達可能なノードの集合である．そして，対象範囲（TD_Rate）の計算を以下の式で行う．

$$\mathrm{TD\_Rate} = \frac{N(V_C \cup V_I \cup V_A)}{N_G - N_I} \quad (10)$$

ここで N_Gはシステム全体のノード数，$N(V)$はノード集合 Vのノード数，N_Iはインターネットモジュールのノード数を示す．TD_Rate により脆弱性のネットワークシステムへの影響度が一意に決定できることになり，細かい影響度の利用が可能である．TD_Rate の算出は従来の CVSS 利用にも利点がある．TD_Rate の値を TD の「利用環境の範囲」と照合することで既存 CVSS が利用可能であることに加え,利用者ごとの入力曖昧さを排除することが可能となり，環境値評価の一意性や再現性を提供することが可能になる．

3.5 まとめ

本章では，複数機器・複数機能の相互接続により構成されるネットワークシステムに対し，機器の機能特徴を失うことなく表現可能なマルチレイヤネットワークモデルを提案した．また提案ネットワークモデルを利用し，標的型攻撃に対する安全なネットワークシステムの設計方法論を構築するための定量尺度として，脆弱性影響度の定量尺度を提案した．

提案ネットワークモデルはこれまでのグラフ理論によるグラフ表現にレイヤ構造を取り入れ拡張したモデルであるが，ノード，リンクはそれぞれ単一種類ではなく複数の種類を持つ．そしてそれら複数種類のノード，リンクの組み合わせによりこれまでのネットワークモデルでは表現が困難であったスイッチやルータ，ファイアウォールなどの各ネットワーク機器を，機能特性を保ったまま表現可能にし，仮想化環境への対応も可能にした．

またリンクとノードの集合と，リンク種別やノード種別などの各属性を写像

として表現することでモデルの定式化を行った．集合の操作を行うことで抽出される各レイヤのネットワークは，単一レイヤ・単一種類ノードで実現される既存のネットワーク設計理論における種々の手法を適用することも可能である．

一方，最適性を示す尺度がマルチレイヤのモデルに適していないために，既存ネットワーク設計理論は提案ネットワークモデルへの直接適用が困難である．そこで，提案ネットワークモデルを用いた定量尺度として，脆弱性影響度の提案を行った．本章で提案したモデルとモデル評価尺度を用いることで，マルチレイヤネットワークモデルにおける標的型攻撃対策の安全性設計とその評価を検討することが可能となった．

参考文献

[１] David M. Nicol, Jason Liu, Michael Liljenstam, Guanhua Yan: Simulation of large scale networks I: simulation of large-scale networks using SSF, Proceedings of the 35th conference on Winter simulation, pp.650-657 (2003)

[２] M. Bakhouya and J. Gaber and A. Koukam: Immune-Based Middleware for Large Scale Network, Annual IEEE Conference on Local Computer Networks, pp.230 (2002)

[３] Fabian Fischer, Florian Mansmann, Daniel A. Keim, Stephan Pietzko and Marcel Waldvoge: Large-Scale Network Monitoring for Visual Analysis of Attacks, Proceedings of the 5th international workshop on Visualization for Computer Security, pp.111-118 (2008)

[４] P. Belotti, F. Malucelli, and L. Brunetta: Multicommodity network design with discrete node costs, Networks, vol.49, issue 1, pp.90-99 (2007)

[５] C. Chekuri, F. B. Shepherd, G. Oriolo, and M. G. Scutella: Hardness of robust network design. Networks, vol. 50, issue 1, pp.50-54 (2007)

[６] El-Sayed M. El-Alfy: Applications of genetic algorithms to optimal multilevel design of MPLS-based networks, Computer Communications, vol. 30, issue 9, pp.2010-2020 (2007)

[７] Hu-Gon Kim, Chun-Hyun Paik, and Yong-Joo Chung: Heuristics for the Access Network Design Problem in 3G Mobile Communication Networks Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control, (2008)

[８] Eric Rosenberg: Hierarchical topological network design, IEEE/ACM Transactions on Network, vol.13, issue 6, pp.1402-1409(2005)

[９] Pietro Belotti, Antonio Capone, Giuliana Carello, and Federico Malucelli: Multi-layer MPLS network design: The impact of statistical multiplexing, Computer Networks, vol. 52, issue 6, pp.1291-1307(2008)

[１０] Freek Dijkstra, Bert Andree, Karst Koymans, Jeroen van der Ham, Paola Grosso, and Cees de Laat: A multi-layer network model based on ITU-T G.805 Computer Networks, vol.52, issue 10, pp.927-1937(2008)

[１１] Paulo Salvador, Antonio Nogueira, and Rui Valadas: Local Area Network Modeling for Performance Prediction , Proceedings of the 32nd IEEE Conference on Local Computer Networks, pp.249-251, 2007.

[１２] Sami J. Habib: Redesigning network topology with technology considerations, International Journal of Network Management, vol.18, issue 1, pp.1-13 (2008)

[１３] 片岡直登：ネットワーク設計問題，朝倉書店 (2008) .

［１４］P. Mell, K. Scarfone, S. Romanosky: A Complete Guide to the Common Vulnerability Scoring System Version 2.0, http://www.first.org/cvss/cvss-guide.pdf (2007)
［１５］情報処理推進機構：セキュリティセンター:共通脆弱性評価システム CVSS v2 概説 , 情報処理推進機構（オンライン), 入手先 <http://www.ipa.go.jp/security/vuln/SeverityCVSS2.html> （参照 2014-10-29)
［１６］National Institute of Standards and Technology: National Vulnerability Database, NIST(online), available from <https://nvd.nist.gov/> (accessed 2014-12-23)
［１７］情報処理推進機構：JVN iPedia, 情報処理推進機構（オンライン), 入手先 <http://jvndb.jvn.jp> （参照 2014-10-29)

第4章 ネットワークシステム設計における標的型攻撃シミュレーション

4.1 背景

本章では，ネットワークシステム設計時における標的型攻撃対策の検証手法として，標的型攻撃のシミュレーションを提案する．設計時における攻撃シミュレーションを行うため，ネットワークシステムのモデリング，および標的型攻撃のモデリングを行うこととし，そのためのデータモデルの提案を行う．

序論で述べたように，本研究では標的型攻撃対策として大きく２つの方法を提案している．一つは内部ネットワークでも，攻撃の検出や対策が容易に行えるようにシステムを設計すること，もう一つは，通常利用される通信を模倣して攻撃を隠ぺいしている遠隔操作通信を検出することである．本章では，前者の対策を提案する．

本章で提案する設計検証手法は，標的型攻撃における第4段階（基盤構築）の後の攻撃行動に着目したものである．序章で述べたとおり，標的型攻撃は執拗でかつ多くの攻撃手法が使用されるため，その通信パターンは複雑なものとなっている．そのため，標的型攻撃による侵入後の影響を調査することは容易ではないが，標的型攻撃からネットワークシステムを防御するためには，内部ネットワークへ侵入された後に，どこにどのような影響があるかを検討し調査することが重要である．しかし，ネットワークシステム上にどのような脆弱性が存在し，どのような対策が効果的かを調査検討するために，実際の環境を構築し，標的型攻撃の不正プログラムを動作させることで現実的ではない．よって，設計情報を元に攻撃をシミュレーションするという手法を用いて，標的型攻撃の影響がどのように及ぶかを調査することは対策の検討に有益である．

シミュレーションを行うために，まず，様々なネットワーク機器を設計上で表現しなければならない．機器にはウェブサーバ，メールサーバ，ネームサーバ，ディレクトリサーバ，プロキシサーバといったサーバ機器，また，スイッチ，ルータ，ファイアウォール，ロードバランサ，などのネットワーク機器を含む必要がある．本研究では，それらを定式的に記述可能とするものとする．なお，データモデルそのものはネットワークシステムの設計情報であるため，モデル内でネットワークの時間的な変化は考慮しないものとする．さらに，ネットワークシステム上で動作する標的型攻撃で用いられる不正プログラムの挙動をドメイン記述言語（以下DSL）として定義し，DSLに基づいてネットワークシステム上の不正プログラム動作をシミュレートするプログラムを開発する．

以下より，不正プログラム動作のシミュレーションを行うことを「脅威トレース」と呼び，シミュレーションを行うためのプログラムを「脅威トレーサ」と呼ぶ．脅威トレーサ本体はScala[1]を使用して実装を行い，ネットワークシ

ステムのモデルと標的型攻撃のモデルを合わせて動作させることでシミュレーションを可能とする．設計者が机上で設計，シミュレートする既存の手法と提案手法を比較して，提案手法は網羅性および再現性において有効であると考えられる．提案手法は設計作業者の負荷を軽減し，より厳密なネットワークシステムの標的型攻撃対策設計検証が可能となる．

4.2 関連研究

ネットワークのモデリングについては，第3章で提案したネットワークモデルをベースとする．よってネットワークのモデリングについての関連研究は3.2章に記載したものが参照可能であるため，ここでは記載しないこととする．

標的型攻撃をモデリングするにあたっては，攻撃段階の表現，および，攻撃に用いられる不正プログラム動作を考慮する必要がある．不正プログラムのモデリングについてはFred Cohenによるコンピュータウイルスの定義が挙げられる[2]．Cohenはコンピュータウイルスの特徴を定義し，プログラムがどのような動作を行えばウイルスとするかを最初に定義した．Gregoire JacobらはCohenの定義を元に，MyDoomなどの旧来型不正プログラムの振る舞いに関する記述方法を提案している[3]．R.K.Shyamasundarらは不正プログラムの内部的なふるまいと外部的な振る舞いの両方に着目し，信頼できるプログラムとの挙動の違いから不正プログラムを検出するアプローチを提案している[4]．これらはいずれも不正プログラムそのものの挙動を定義することに主眼が置かれており，個々の不正プログラムを定義するには有効である．しかし，標的型攻撃のように複数の不正プログラムの使用や多段階の攻撃スキームを定義するものではない．また，現在米国が進めている情報交換スキームのSCAP[5]ファミリーでも不正プログラムの記述方法であるMAEC（Malware attribute enumeration and characterization）[6]が検討されている．MAECは属性ベースの不正プログラム記述言語であり，MITRE[7]により開発，維持がなされている．MAECは不正プログラム情報の形式的なやり取りに適しており，様々な種類の不正プログラムにおいて，その構造や詳細な振る舞いを記述できる．MAECは不正プログラムを列挙するために3層の抽象モデルを定義している．第一層として不正プログラムの能力や振る舞いの記述，第2層で不正プログラムの特徴情報（ファイル名など），第3層では特徴情報をまとめる記述が可能となっている．MAECは不正プログラムのデータベースを作成し，情報交換を行うために作られているため，不正プログラムの分類，記述に適した言語である．しかし，MAECは不正プログラム単体の構造を記述しており，ネットワーク上でどのような通信を行うかといった情報や，C&Cからコントロールされて複数の不正プログラムが通信しあうことで起きる事象などを記述するためのものではない．本章ではネットワー

クにおける標的型攻撃のふるまいをシミュレーションすることを目的としており，不正プログラムを一意に特定するために考えられたMAECはの利用は適切ではないと考えられる．以上を勘案し，ネットワーク上の動作に特化した不正プログラムのモデルを新たに定義することとした．

4.3 脅威トレーサのフレームワーク

4.3.1 脅威トレーサの基本アイデア

脅威トレーサはネットワークシステムと不正プログラムの挙動を解析し，不正プログラムの動作をシミュレーションするものとする．不正プログラムは初期潜入後にバックドアを開設しC&Cサーバと通信を行う．その後，攻撃に必要なネットワークサービスを動作させたり，感染動作を広げたりするなどして，次の段階に攻撃を進行させる．このように，不正プログラムが侵入し，何らかの活動を行うことによって，ネットワークのサービスやアクセス制御の状態が変化していくと考えられる．このような動作パターンは不正プログラムの挙動によって様々に表れ，事象として独立している．これらは並列的，段階的に動作しており，そのシミュレーションを行うことで時間的な変化を表すことが可能となる．以下にこの考え方を設計時のシミュレーションに適用することを検討する．

不正プログラムが動作することで，設計上考慮されていない端末の通信が行われる，もしくは設計上考慮されていないプロトコルで通信が行われるといったことが発生した場合，それはネットワークモデル上では，元の設計情報に対してアクセス経路の追加やサービスノードの追加が行われ，ネットワークモデル情報に変更が加わるということと同意と考えられる．不正プログラムの動作によって変更される情報は様々異なるため，不正プログラム動作の違いが複数のネットワークモデル情報として表現されることとなる．さらに次の不正プログラム動作が発生すると，事前の動作によって表現されたネットワークモデル情報を元として，アクセス経路追加やノード追加などが行われる，この動作を繰り返すことで攻撃が段階的に進行する状況を，設計情報におけるアクセス状態やノード状態の変化として網羅的にシミュレートすることが可能となる．

以上．シミュレーション動作の概念を図 4.1に示す．図では最初に設計されたネットワークシステムモデル（状態A）に攻撃活動のモデルを適用することで，複数のネットワークモデルが生成され（状態B1～B3），さらにそれらに次の攻撃活動が適用されることで次のネットワークモデルが生成される（状態C1～C5）ことを表している．

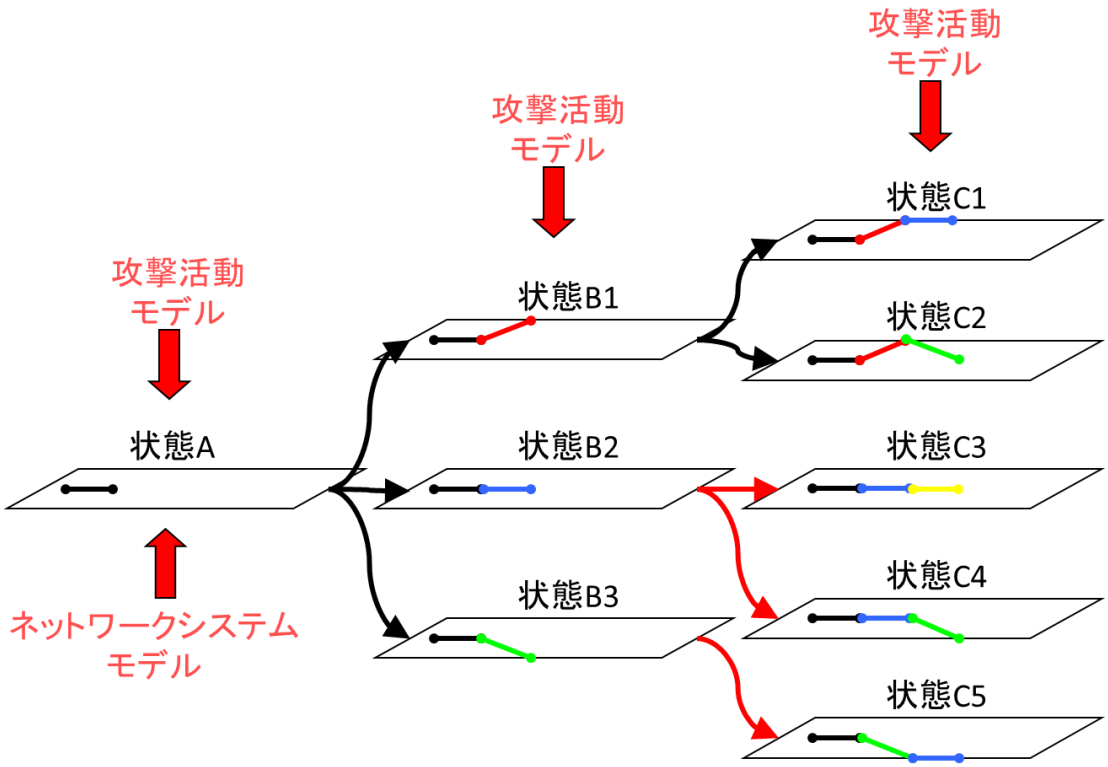

図 4.1 シミュレーションの概念図

4.3.2 ネットワークモデル上での不正プログラムの振る舞い

標的型攻撃は目的を達成するために複数の段階を経て行われるが，システム設計の観点からその動きを大別すると，初期の侵入とその後に分かれる[8]．初期の侵入方法としては，不正プログラムが添付された標的型メールが送付される以外にも，ブルートフォース攻撃が行われたり，その他ソーシャルエンジニアリング的な攻撃が行われたりと様々である．初期の侵入対策としては，アンチウイルスソフトウェアや侵入検知，ソーシャルな攻撃に対する社員へのセキュリティ教育など，様々な対策を行うことが考えられるが本章の範囲ではないため，初期潜入対策は考慮しない．本章では侵入後の不正プログラムにおけるネットワークシステム内の攻撃活動を対象とする．内部ネットワークへ侵入後の不正プログラム動作をシミュレーションすることで，ネットワークシステム設計時において，網羅的な標的型攻撃対策を考慮することが可能となると考えられる．

第1章で述べたように攻撃者は内部ネットワークへ侵入後，複数の端末に攻撃を行い，バックドアを動作させることで攻撃基盤を構築する．そのような攻撃基盤を確立する際の詳細な動作を改めて下記に記載する．

１）不正プログラムはバックドアを利用して攻撃者が操作するC&Cサーバと通信を行う．
２）ネットワークシステム内の設定情報や構成情報を収集する．
３）収集した情報を元に，ネットワークシステム内の別の端末に対して不正プログラムを拡散させる．
４）アカウント情報を入手した場合はそのアカウント情報を使用して正規アカウントとしてログインを行い，さらに情報を収集する．

このようにして，多数の端末に侵入することで攻撃者は攻撃基盤を強固なものとし，それを足掛かりにして目的の情報を持った機器に到達する．最終目標として，秘密情報の窃取，Webサイトの改ざん，他のサイトへのさらなる攻撃といったことを行う．脅威トレーサではこれらの挙動をネットワークモデル上に再現できるようにする．

4.3.3 タスクスケジュール法による並列分散処理

脅威トレース計算の複雑さはネットワークシステムのノード数，リンク数，および不正プログラムが持つ機能の多さに依存している．加えて，脅威トレースはネットワークシステム上で不正プログラムが複数同時に動作することも考慮しなければならない．第3章の提案ネットワークモデルでは，ネットワークモジュールとして，PC，ルータ，スイッチ，ロードバランサ，ファイアウォール，IDS，アンチウイルスアプライアンス，各種サーバなどが定義可能である．しかしマルチレイヤネットワーク構造を解析すると，物理リンクはまばらであったとしても，アクセス制御されていない論理ネットワーク構造が非常に密なリンクとなることがあり，論理リンクの密度は物理リンクのトポロジに依存しない．不正プログラムの攻撃活動をシミュレートするには，論理ネットワークのアクセス制御構造を考慮する必要があり，また，大企業のネットワークや政府系のシステムのような，巨大のネットワークシステムでは複数の不正プログラムの動作によって侵入が行われることを想定しなければならず，そのような場合はシミュレーション時間が非常に長くなることが予想される．その計算は高性能な機器を使用するか，複数の機器で並列分散処理を行うことが望ましい．そこで，ここでは並列分散処理による高速化について検討する．

計算が並列分散処理で複数のCPUに振り分けられることによって，タスクスケジュール法を使った様々なテクニックが利用可能となる．提案ネットワークモデルにおいて，個々のノードvをタスクとし，個々のリンクeをタスクの依存関係だとすると，モデルは相互依存したタスクインタラクショングラフと考えることができる．タスクインタラクショングラフを表すネットワークモデルは

タスクフローグラフに変換可能である．この変換処理はプログラムをコンパイルすることと同様であり，自動的に変換可能であることを意味する．タスクセットのフローを明示的に割り当てることで，スケジューラの先読みを減らし，計算性能を高めることができる．しかし，ネットワーク設定はシステム更新等で変更された場合は，再度，タスクフローグラフを再計算しなければならない．動的な脅威トレースはタスクスケジューラによる計算と動的なPC管理をもとにしたタスク割り当てが必要となる．そこでタスクスケジューラは静的に計算されたタスクフローグラフを参照することとする．このようにすることで，脅威トレースを効率的に並列分散処理することができる．

4.3.4 ドメイン記述言語

ネットワークシステムで動作する不正プログラムをシミュレーション可能な形で記述するために，提案ネットワークモデルと不正プログラムの動作を合理的に記述できるDSLを作成した．例として，提案ネットワークモデルの各レイヤで記述される情報を元に，ルータでつながった2台のPCを接続した構成を図4.2に記述し，その図をDSLで表現したものを図 4.3に示す．PC1とPC2が，NetworkCardとrtable1を持つL3Rモジュールであるrouter1でつながっており，PC1がつながっているネットワークは192.168.0/24，PC2がつながっているネットワークは192.168.1/24であることを記述している．さらに，PC1はTCPポートの22と80，PC2はTCPポートの22,80,443をネットワークサービスとして動作させている．また，PC1が不正プログラムに感染していることを記述している．

次に不正プログラムが攻撃基盤を構築する動作をDSLで記述したものを図4.4に記す．イニシャライズセクションはPCが感染した場合の初期動作を，コントロールセクションはC&Cサーバとのコントロールメッセージの動作を記述している．ここではまず，感染するとバックドアを作成し，C&Cサーバと通信を行い，その後，コントロールメッセージで，不正プログラムが発動，NW情報の収集，C&Cサーバと通信，不正プログラム自身のアップデート， PC2へ侵入可能であれば侵入する，という流れを記述している．

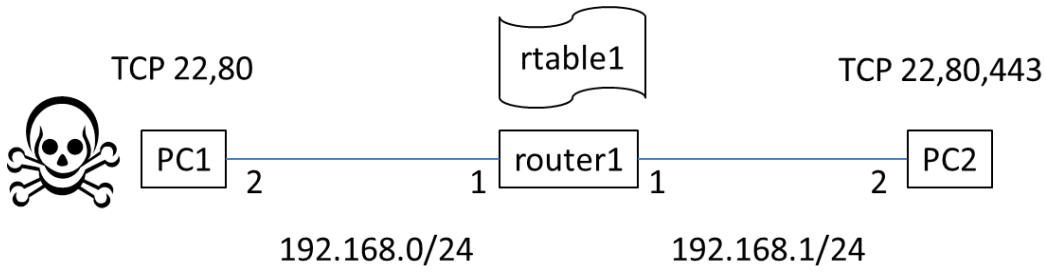

図 4.2 サンプルネットワーク

```
object Sample extends MalwareSimulationLibrary {
        val pc1, pc2 = PCterminal
        val router1 = Router
        val malware1 = Conficker
        val rtable1 = RoutingTable
        pc1 has a networkCard  ‘ ‘192.168.0.2/24’ ’
        pc2 has a networkCard  ‘ ‘192.168.1.2/24’ ’
        pc1 opens tcpPort (22, 80)
        pc2 opens tcpPort (22, 80, 443)
        router1 has a networkCard  ‘ ‘192.168.0.1/24’ ’
        router1 has a networkCard  ‘ ‘192.168.1.1/24’ ’
        router1 has a rtable1
        pc1 connects to router1
        pc2 connects to router1
        pc1 is infectedWith malware1
        run
}
```

図 4.3 DSLで記述したサンプルネットワーク

```
class Sample extends SimpleMalwareLibrary {
        initialize {
                makeBackDoor()
                communicatesWith(cncServer)
        }
        control {
                case Invoke => invokeMalwareCode()
                case GetNetInfo => getsNetworkInfo()
                case HttpMsgCom => communicatesWith(cncServer)
                case Updates => updatesMyselfBy(cncServer)
                case Infection => infectsTo(pc2)
        }
}
```

図 4.4 DSL で記述した攻撃基盤構築

DSLを使用し，このように記述することで，ネットワークシステムにおける標的型攻撃動作のシミュレーションが可能となると考えられる.

4.4 試作システム

このようなシミュレーションが実際に可能であるかを検証するため，脅威トレーサの試作を行った．脅威トレーサの開発にはScalaを使用した．ScalaはDSL記述に適しており，XML等を使ったネットワークシステムや不正プログラム活動を記述することにも適している．また，様々な機能を持ったネットワークシステム（例えば，ルーティング機能，ソフトウェアの動作など）や不正プログラムの振る舞いをシミュレートする場合において，Scalaのトレイト機能を利用することによって効率よくオブジェクトを生成することが可能で，巨大かつ，複雑な並列計算ソフトウェアを容易に実装することが可能である．加えてScalaはJavaライブラリを活用可能で性能の向上も期待できると考えられる．

4.4.1 ネットワークシステムの内部記述

脅威トレーサで扱う提案ネットワークモデルは複数のノード属性を持つことが可能である．ノードとシステム状態に関しての接続状況をシミュレートするためにシステム内部でモジュールの属性を定義している．以下に各属性について説明を行う．

１）Basic module

このモジュールはシステムの基本となるモジュール情報を保持する．アクセッサメソッドは提案ネットワークモデルをベースとしてシーケンシャルナンバー，IDラベル，レイヤ，ノード種別，MACアドレス，IPアドレス，TCPポート，UDPポート，サービス情報などが実装されている．

２）Infected module

このモジュールは不正プログラムの感染をシミュレートし，不正プログラム感染に関するメソッドを持つ．実装されているメソッドは，不正プログラム感染（getsInfectWith），不正プログラム駆除（removeMalware），不正プログラム発動（invokeMalware）である．

３）Network module

このモジュールは提案ネットワークモデルのL3R（IP層）以上の振る舞いをシミュレートし，ネットワークインターフェースと接続ノードの一覧，デフォルトルート，ルーティングテーブル，ファイアウォールルールを持つ．

４）Communication module

このモジュールは他のCommunicationモジュールとの接続を確認し，接続されていればメッセージを送信する．なお，このモジュールはscala.actors.Actorの継承により実装されている．

4.5 攻撃シミュレーション

4.3節で述べたように，攻撃シミュレーションは離散イベントであるため，水平分散イベントシミュレータの実装を行った．DSLの検証や分散イベントシミュレーション実装を検証するため，シンプルなネットワーク構成と攻撃活動をDSLで記述し，動作テストを行うこととした．図 4.5に検証に使用したネットワークシステム概念図を示す．なお，シミュレーションには攻撃動作に関連する機器（図中矢印部分）のみDSLで記述して使用している．PC１は不正プログラムに感染したノードである．図 4.6はシステムの初期状態を，図 4.7はPC1からPC2への不正プログラム観戦活動を，図 4.8はPC1およびPC2から同時にPC3への感染活動が行われていることをシミュレーションしている．以下にそれぞれの内容について説明する．

１）初期状態では，システムはクライアントモジュールとしてPC1,PC2,PC3を，ルータモジュールとしてRouter1,Router2を持ち，PC1が感染しているとする．図 4.6にシミュレータで動作させた結果を記す．ここでMalware(PCTerminal1(pc1))と表示されているものが，不正プログラムが感染していることを表している．

２）PC1の不正プログラムは次の感染ノードを探す．不正プログラムは感染可能なノードとしてPC2を検出する．その後PC2が感染する．図 4.7にある，PCTerminal(pc2):call isInfectable trueが動作となり，その結果がprint以下にlogとして記録されている．Malware(PCTerminal(pc2))とあるように，PC2が新たに感染していることがわかる．

３）PC1とPC2の不正プログラムは同時に次の感染ノードを探す．図 4.8のMalware(PCTerminal(pc1)):action および Malware(PCTerminal(pc2)):actionが同時に起動していることを表す．PCTerminal(pc3):call isInfectable trueとあるとおり，PC3が感染可能であることが個別に検出され，PC3はMalware(PCTerminal(pc2))となっているとおり，不正プログラムに感染する．

このように，データモデル化されたネットワーク設計情報と不正プログラムのネットワーク動作モデルを使用することによって，標的型攻撃で行われるよ

うな段階的な攻撃の進行や，複数の攻撃イベントが同時に発生した場合の攻撃シミュレーションが設計時に可能となることを示した．シミュレーションを行うことで標的型攻撃が内部ネットワークに侵入した場合の攻撃到達性も網羅的に確認行えるようになり，設計漏れなどの発生を低減できることが期待できると考えられる．

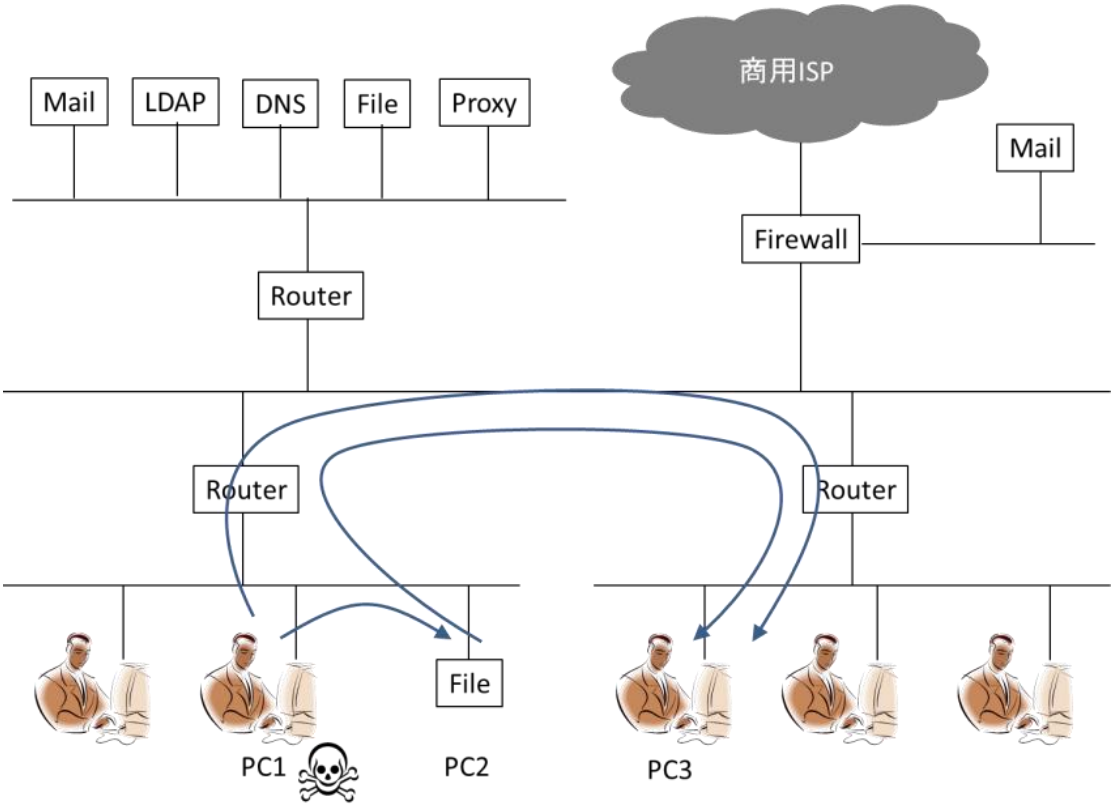

図 4.5 テスト用ネットワークシステム

4.6 結論

標的型攻撃は複数の段階を経て，ネットワークシステム内で探索行為や感染活動が行われる．さらに，攻撃手法は動的に変化するため，対策検討が困難となっている．加えて，近年のネットワークシステムは機能やノードが増加することで構造が複雑化しているため，標的型攻撃を防御するためのネットワークシステム設計はコンピュータによる支援が有効である．本章では標的型攻撃を，DSLを使ってモデリングした．加えて，第3章で提案したネットワークモデルを応用することで，内部ネットワークに侵入後の標的型攻撃の振る舞いをシミュレーション可能であることを示した．結果として，標的型攻撃の攻撃段階が進行していく状況を計算によって明らかにし，ネットワークシステム内の安全性設計が適切であるかどうかを検証できることを示した．

```
$ java -jar MalwareSimulator.jar
PCTerminal(pc1)
PCTerminal(pc2)
Router(router1)
PCTerminal(pc3)
Router(router2)
Malware(PCTerminal(pc1))
```

図 4.6 初期状態のテスト

```
Malware(PCTerminal(pc1)):action
PCTerminal(pc1):send Router(router1)
Router(router1):send PCTerminal(pc1)
PCTerminal(pc1):send Router(router1)
Router(router1):send PCTerminal(pc2)
PCTerminal(pc2):send Router(router1)
Router(router1):send PCTerminal(pc1)
PCTerminal(pc2):call isInfectable true
print
PCTerminal(pc1)
PCTerminal(pc2)
Router(router1)
PCTerminal(pc3)
Router(router2)
Malware(PCTerminal(pc1))
Malware(PCTerminal(pc2))
```

図 4.7 PC2 感染後の状況

```
Malware(PCTerminal(pc1)):action
Malware(PCTerminal(pc2)):action
PCTerminal(pc2):send Router(router1)
....cut...
Router(router1):send PCTerminal(pc2)
PCTerminal(pc2):call isInfectable false
PCTerminal(pc1):send Router(router1)
PCTerminal(pc1):call isInfectable false
Router(router1):send Router(router2)
....cut...
Router(router1):send Router(router2)
PCTerminal(pc3):call isInfectable true
Router(router2):send PCTerminal(pc3)
PCTerminal(pc3):send Router(router2)
Router(router2):send Router(router1)
Router(router1):send PCTerminal(pc2)
PCTerminal(pc3):call isInfectable false
print
PCTerminal(pc1)
PCTerminal(pc2)
Router(router1)
PCTerminal(pc3)
Router(router2)
Malware(PCTerminal(pc1))
Malware(PCTerminal(pc2))
Malware(PCTerminal(pc3))
```

図 4.8 PC3 感染後の状況

参考文献

[１] EPFL: The Scala Programming Language, Typesafe, Inc. (online), available from <http://www.scala-lang.org> (accessed 2014-10-29)
[２] Fred Cohen: Computer Viruses Theory and Experiments, Computers & Security 6, pp. 22-35 (1987)
[３]Jacob, Grégoire and Filiol, Eric and Debar, Hervé: Malware as interaction machines: a new framework for behavior modelling, Journal in Computer Virology 4(3), pp. 235-250 (2008)
[４]Shyamasundar, R.K. and Shah, Harshit and Kumar, N.V.Narendra: Malware: From Modelling to Practical Detection, Distributed Computing and Internet Technology vol. 5966, pp. 21-39 (2010)
[５] NIST: The Security Content Automation Protocol(SCAP), National Institute of Standards and Technology, available from <http://scap.nist.gov/> (accessed 2014-10-29)
[６] The MITRE Corporation: Malware Attribute Enumeration and Characterization, The MITRE Corporation(online), available from <http://maec.mitre.org> (accessed 2014-10-29)
[７] The MITRE Corporation, available from <http://www.mitre.org> (accessed 2014-10-29)
[８] 情報処理推進機構：「標的型メール攻撃」対策に向けたシステム設計ガイド，情報処理推進機構（オンライン），入手先 <http://www.ipa.go.jp/files/000033897.pdf> （参照 2014-10-29）

第5章 通信経路上の情報挿入による偽装通信の検出

5.1 はじめに

本章では，標的型攻撃の遠隔操作で使われるバックドア通信を迅速に検出することを目的とし，クライアントとサーバの通信を中継している装置で，通信に特定の情報を付加し，その情報がどのようにクライアントで処理されるかを確認することで，バックドア通信かどうかを判別する方法を提案する．

具体的な手法としては，HTTP プロキシサーバで本来の通信には存在しない Cookie ヘッダを付加し，HTTP クライアントがその Cookie ヘッダを正しく処理した場合は正規のブラウザ，Cookie ヘッダに応答しないなど，正しく処理しない場合はバックドアとして検出する．HTTP プロキシサーバで Cookie ヘッダの応答を確認するためには，クライアント毎の Cookie ヘッダ発行状況を管理する必要があるため，本来の通信には存在しない Cookie ヘッダを生成し，その発行状態を管理する HTTP プロキシサーバを実装する．さらに，標的型攻撃で使用されたバックドアが動作可能な環境を構築し，実際にバックドアを動作させる．それによって本提案方式の有効性を評価し，本提案方式でバックドアによる HTTP 通信をリアルタイムで検出可能なことを確認する．加えて，本提案方式の優位性，および検出が行われない条件等に関して考察を行う．

以下，本章における標的型攻撃の着眼点について 5.2 節で概説し，5.3 節で標的型攻撃の詳細動作を述べ，5.4 節で関連研究について記載する．5.5 節では提案手法を，5.6 節では提案手法の実装をそれぞれ記述し，5.7 節で実際の標的型攻撃サンプルを使った評価実験とその結果を示す．5.8 節で考察および課題を検討し，最後に 5.9 節で本章をまとめることとする．

5.2 本章における標的型攻撃の着眼点

標的型攻撃の全体像は過去の事例を基に，情報処理推進機構（以下 IPA）[1]や Aditya K Sood[2]らが分析を行っている．

標的型攻撃はある特定の攻撃方法を指し示すものではない．標的型攻撃とは，標的となる情報を持つ人や組織だけを狙い，情報を窃取する攻撃である．複数の攻撃段階があり，段階に応じた攻撃手法が使われ，最終的な目的を遂行する．そのため，標的型攻撃対策を検討する際には，どの段階のどの攻撃手法へ適用する対策であるかを明確にすることが重要である．

序章で述べたとおり，攻撃は複数の段階を経て行われる．改めて下記に本章の内容を検討するうえで重要となる攻撃段階について，概要を記す．

・計画立案：攻撃目標の設定，関連情報の調査が行われる．

・攻撃準備：特定の個人や組織に向けた標的型メールやバックドアを管理制御するための C&C サーバが準備される．
・初期潜入：標的型メールが送付され，不正プログラムが実行される．
・基盤構築：バックドア開設，端末情報入手，構成情報入手等が行われる．
・内部侵入・調査：他端末侵入，サーバ侵入，管理者情報窃取等が行われる．
・目的遂行：情報窃取，システム破壊等が行われる．
・再侵入：再度侵入が行われる．

計画立案，攻撃準備の段階については外部ネットワークで行われるため，内部ネットワークでの対策範囲ではない．内部ネットワークでの対策でカバーできるのは，初期潜入以降となる．初期潜入対策としては，ファイアウォールやアンチウイルスといった，ネットワークの入口での対策が行われるが，第 2 章の議論から，標的型攻撃は巧みにそれらの防御を突破して行われることを想定しなければならない．初期潜入が成功すると，攻撃者は遠隔操作のためのバックドアを開設し，次の攻撃段階に移るための情報を入手する．攻撃の段階を進めることなく，被害を最小限にするためには攻撃を早期に検出することが重要である．そこで本章では初期潜入後の基盤構築段階で使用されるバックドア通信に焦点をあてて対策を検討することとする．

5.3 攻撃通信仕様詳細

ここでは，初期潜入から基盤構築を経て情報窃取に至るまでのネットワーク上の動作を，図 5.1 を用いて説明を行う．

１）初期潜入段階

まず，攻撃者は不正プログラムが動作するように細工された pdf などのファイルを添付メールで送信する．本文には受信者が添付ファイルを開封することを誘う内容を記載する．受信者はその添付ファイルを開封することで，不正プログラムに感染する（図 5.1 (1) (2)）．感染後，不正プログラムは C&C サーバより追加の不正プログラムをダウンロードするなどして，内部ネットワークのクライアントを遠隔操作するためのバックドアプログラムを動作させる（図 5.1 (3) (4)）．

２）基盤構築段階

バックドアプログラムは，内部ネットワークから外部ネットワークの攻撃者に向かって，通信パスを確立する．これにより，ファイアウォールなどの境界防御を通過することが可能となる（図 5.1 (5) (6)）．

３）内部侵入，目的遂行段階

攻撃者はこの通信パスを利用し，攻撃コマンドを送信することで遠隔操作を行い，情報を窃取する（図 5.1（7）（8）（9））.

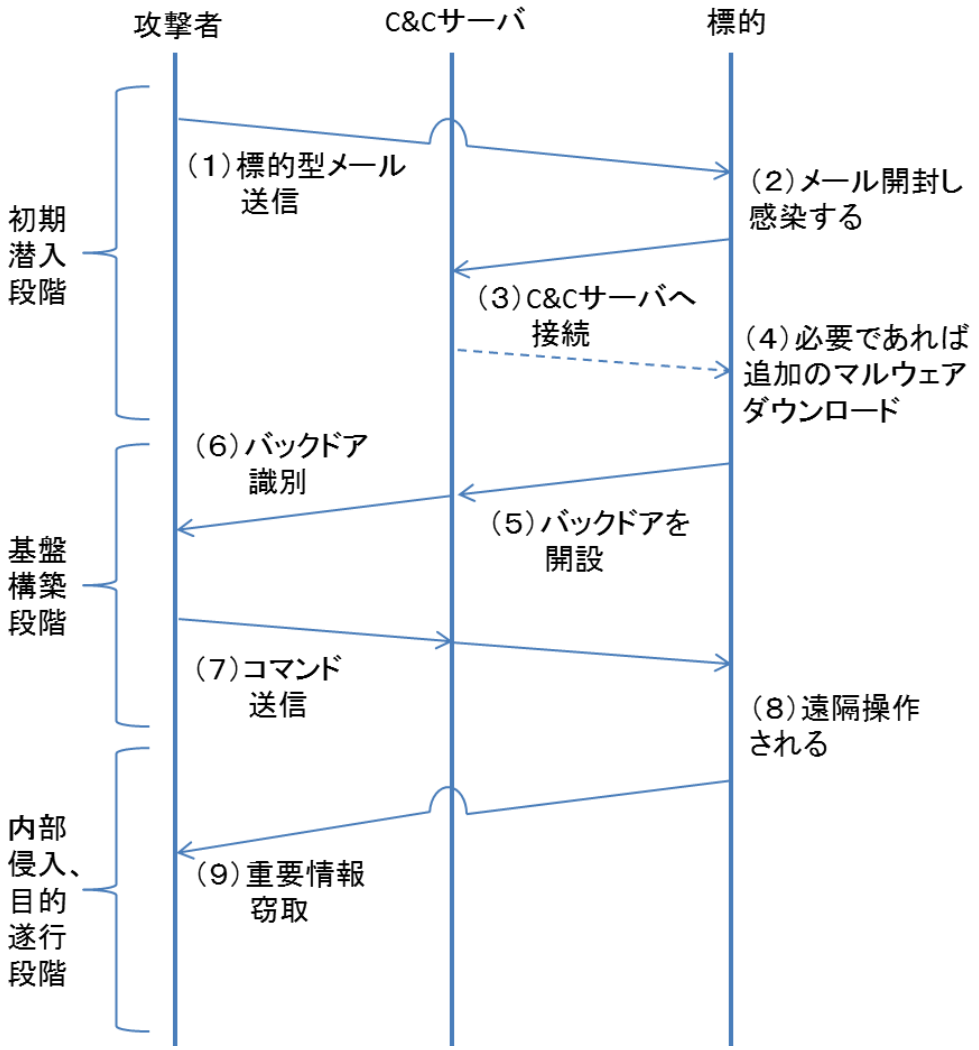

図 5.1 ネットワーク上での情報窃取動作

第 1 章で述べたように，バックドアが内部ネットワークから外部ネットワークへ行う通信パスの種別は，IPA[3]によると 2013 年時点で図 5.2 のとおりである．およそ半数が外部ネットワークへ直接通信を行い，残り半数が HTTP プロキシサーバ（以下プロキシサーバ）を利用する通信となっている．

内部ネットワークから外部ネットワークへの通信に制限がない場合などは直接パスの独自プロトコルも含めてすべてのバックドア通信が成立する．企業など，各端末がインターネットに直接接続する必要が無く，ファイアウォールやプロキシサーバがある場合は，プロキシサーバを経由しない内部ネットワーク

から外部ネットワークへの通信を拒否することにより，プロキシサーバに対応していないバックドア通信は止めることができる．しかし，プロキシサーバに対応しているバックドア通信は，内部ネットワークのクライアントから行われる通常の通信と同等に扱われるため，止めることはできない．一般的な内部ネットワークからの通信とプロトコル上は区別がつかないことが，攻撃の検出をより困難なものとしている．

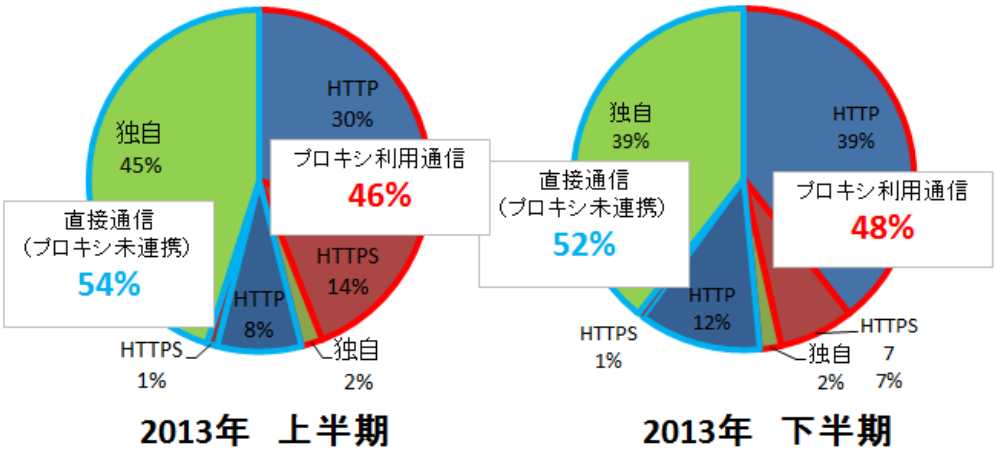

図 5.2 通信パス種別（IPA 資料より引用）

このような通信を行うバックドアを使うことで，攻撃者は気づかれることなく内部ネットワークのクライアントを遠隔操作し，いくつかの攻撃段階を経て情報窃取を行っている．

5.4 関連研究

前述のとおり，内部ネットワークの構成に合わせてプロキシサーバを利用した HTTP 通信を行うバックドアは検出が困難である．本節では既存の対策手法を述べるとともにそれぞれの課題について記載する．標的型攻撃のバックドア通信をリアルタイムに検出する手法について未だ検討の余地があると考えられる．

5.4.1 検出アプローチ

Guido Schwenk らは，HTTP の隠れ通信を検出するために，HTTP ヘッダや本文の長さ，エントロピー量，時間などを利用して，プロキシサーバでアノマリ検出を行う DUMONT システムを提案し，実装を行っている[4]．Yao-jun DING らはパケットの長さや時間間隔といった特徴量を，C4.5 アルゴリズムを使用して分類することによって，HTTP の隠れ通信の検出を試みている[5]．Paul Giura ら

は標的型攻撃が複数の段階にわたり攻撃手法が組み合わされていることに着目して，攻撃の全体像を定義し，その全体像から，複数のフィルタを効果的に組み合わせることで，攻撃検出の精度を上げる手法を提案している[6]．Marco Balduzzi らはユーザのアクセスする URL をクラスタリングし，送信元機器のグルーピング情報と突き合わせることで，標的型攻撃の検出を行う SPuNge システムを提案し，実装を行っている[7]．Johannes de Vries らは標的型攻撃の各段階において，ネットワーク型 IDS やホスト型 IDS などからの情報を収集分析することで，攻撃を検出する考え方について考察を行っている[8]．これらの方法は，正常な通信と不正な通信の閾値の設定が統計量やクラスタリングの誤差に依存するため，誤差の発生が避けられない．また，母集団の大きさや偏りによって検出精度が左右されるなど，統計的手法を使うことに起因する課題が残る．さらに，時間間隔などの統計値を得るためには一定期間の観測が必要であり，検出するまでに時間を要する．ログ解析などでも，事後の解析となるためリアルタイム性に欠ける．加えて，既存のネットワーク構成に対して，これらの対策が容易に適用可能かどうかという，実用面の観点での検討も必要となる．

5.4.2 サンドボックスによる検出

ネットワーク上の装置で HTTP 通信を横取りし，サンドボックス等を動作させることによってバックドア通信の検出を試みる商用製品が存在する[9][10]．これには全ての端末のパケットを取得し，データも含めて内容を確認することが必要となる．しかし，全トラフィックデータとなるとその量は膨大であり，その内容を逐次確認して標的型攻撃を見分けるには時間がかかる．また，標的型攻撃で使用されるバックドアは環境への依存性が高い．実際の環境と異なるサンドボックス上で動作するとは限らず，運用にあたっては攻撃かそうでないかを判断するための技術を持った人員を確保する必要があるなど，大きなコストが必要となる．よって，現時点において本方式が利用可能な組織は限られ，広く一般的に対して適用することが可能とは言い難い．

5.4.3 ネットワーク設計による検出

バックドア通信が容易に成立する原因として，外部ネットワークから内部ネットワークへの通信は制限されるが，内部ネットワークから外部ネットワークへの通信が制限されないことが考えられる．バックドアが外部ネットワークへ直接通信を行うことを防ぐため，内部ネットワークから外部ネットワークへは直接通信ができないようなネットワーク構成にするという対策が考えられる．内部ネットワークからの各種の通信サービス利用については，Web の閲覧はプロキシサーバ，電子メール送受信は内部メールサーバ，名前解決には内部 DNS

サーバといったように，内部専用サーバを用意し，それらのサーバからのみ外部ネットワークとの接続を許可するようにすればよい（図 5.3）．ファイアウォールなどでクライアントからの直接接続を検出することで，攻撃の可能性を知ることができる．

このネットワーク構成は，企業などでは一般的に用いられている構成であり，実用性は高いと考えられる．しかし，バックドアがプロキシサーバを利用するように仕組まれているなど，内部ネットワークの構成に合わせて通信が行われる場合には対応が困難となる．

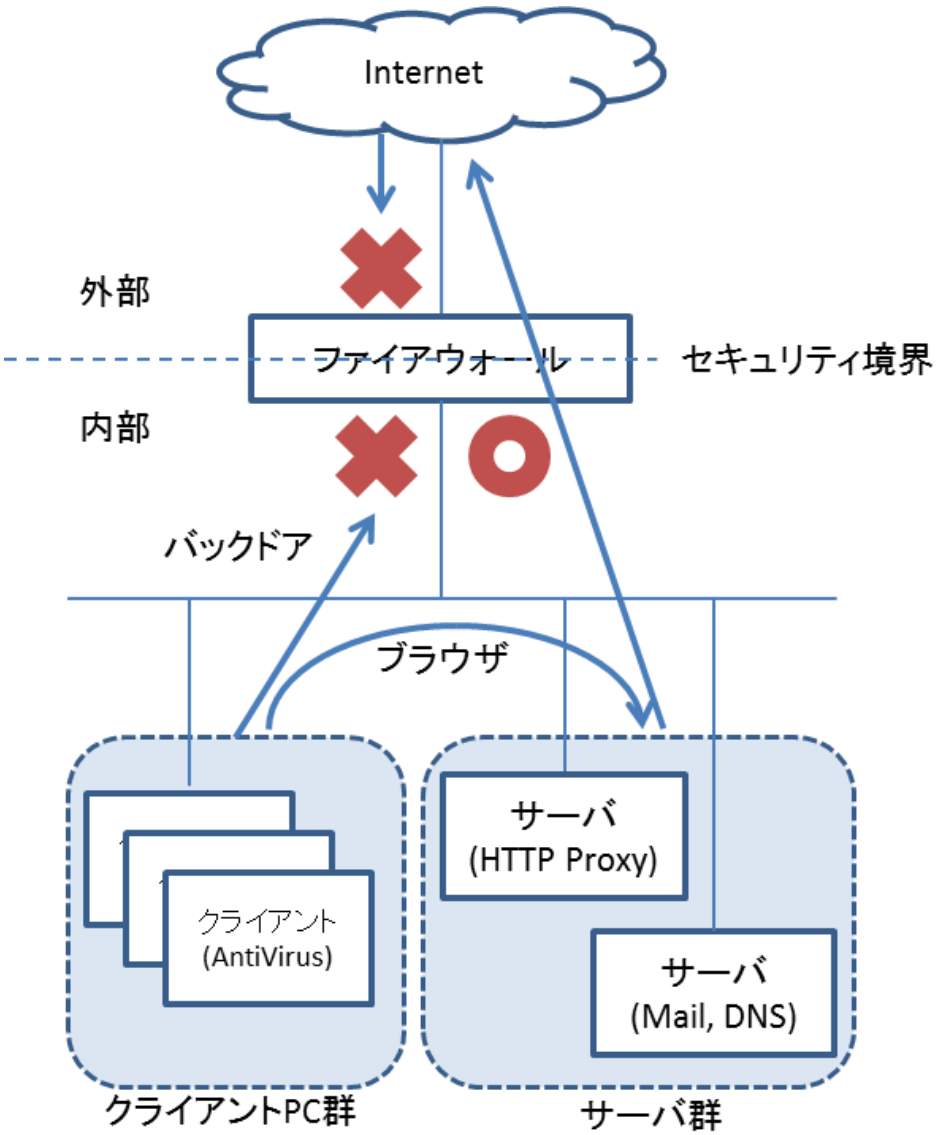

図 5.3 ネットワーク設計による対策

5.5 提案手法

5.4 節で述べたように，HTTP を使用するバックドア通信はリアルタイム検出や誤検出かどうかの検証が容易ではない．さらに，5.3 節で述べたとおり，プロキシサーバに対応しているバックドア通信は，内部ネットワークのクライアントから行われる通常の通信と同等に扱われるため，プロキシサーバやファイアウォールでの検出は困難である．しかし，プロトコル的には正常な通信を装ったとしても，バックドアは遠隔操作プログラムでありブラウザではないため，ブラウザが持つ機能全てを備える必要はなく，機能上の違いが生じる可能性がある．

そこで，本来の通信には存在しない情報をプロキシサーバ上で挿入し，挿入した情報がクライアントでどのように処理されるかを識別し，機能の差を確認することで，リアルタイムにバックドア通信を検出する手法を提案する．大まかな処理の流れを下記に記すとともに，図 5.4 に図示する．

１）クライアントからの通信要求を通信経路上のプロキシサーバで受信する．プロキシサーバは受信した通信を宛先に転送する．
２）宛先からの応答がプロキシサーバに返される．プロキシサーバはその応答に対して，クライアントが処理可能な情報を挿入してクライアントに返す．
３）当該クライアントから次の通信要求が来た際に，挿入した情報の処理結果をプロキシサーバで確認する．挿入した情報に関する通信内容は削除して宛先に送信し，本来の内容を変えることなく通信を継続する．

プロキシサーバで情報を挿入し，クライアントからその処理結果を受信するといった動作を HTTP で行うためには，ヘッダなら Cookie，ETag，コンテンツなら JavaScript といったような，レスポンスのある通信を利用することが考えられる．5.5.1 項では，この動作に適した挿入情報を選定し，具体的な実現手法を 5.5.2 項および 5.5.3 項で記載する．

5.5.1 手法の選定

バックドアとブラウザで違いが出やすい挿入情報を選定するために，Cookie ヘッダ，ETag ヘッダ，JavaScript などのコンテンツ，およびバックドアで処理されない情報についての調査を行った．調査の結果，挿入する情報として Cookie ヘッダを使用することとした．以降にその概要を記す．

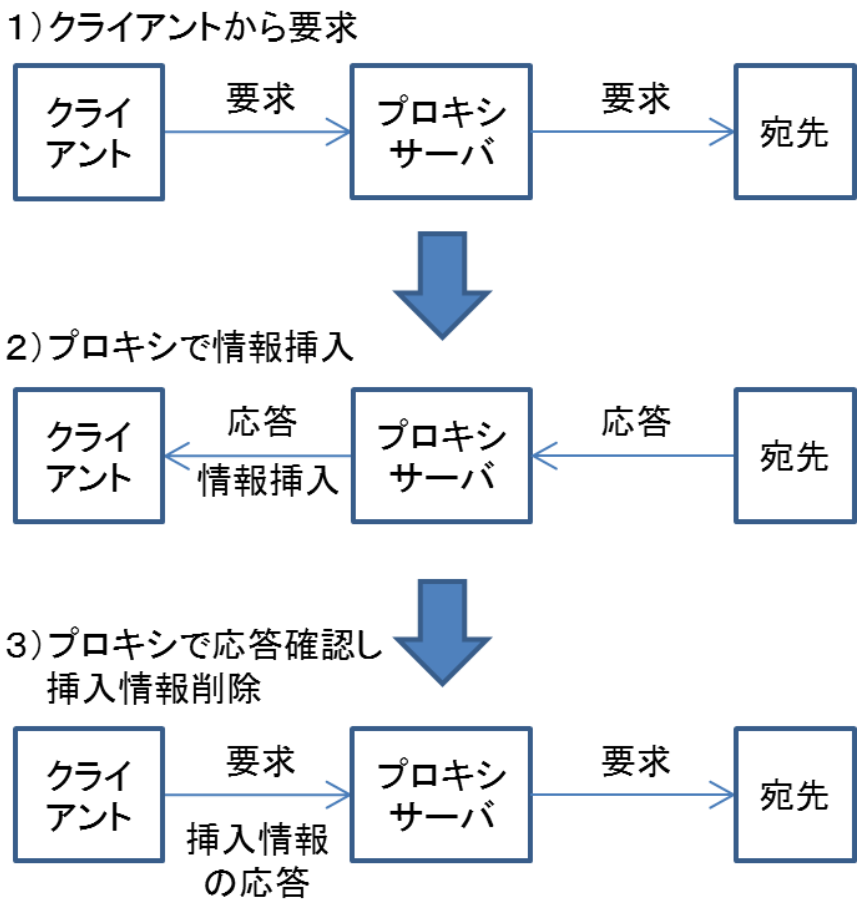

図 5.4 提案手法概要

5.5.1.1 ヘッダ挿入

１） Cookie ヘッダ

通常利用時に Cookie ヘッダがどの程度使用されているかを確認するため，九州工業大学小出研究室のプロキシサーバで HTTP ヘッダ情報を取得しヘッダの出現頻度を調査した．調査条件は表 5.1 の通りである．

調査期間	2013/4/1～2013/6/30
利用者数	15 名
IP アドレス数	45
総リクエスト数	4388740
利用ブラウザ	IE，Firefox，Chrome 等

表 5.1 調査条件

調査の結果，リクエストヘッダの総数は 31886178，種類は 250 種存在した．その一部として，出現頻度上位 10 種のヘッダを表 5.2 に記載する．

リクエストヘッダ	出現頻度
Host	99.97
Connection	98.19
User-Agent	93.99
Accept-Encoding	89.87
Accept	85.75
Accept-Language	77.15
Referer	68.55
Cookie	38.56
Accept-Charset	21.45
X-Dropbox-Locale	10.96

表 5.2 リクエストヘッダと出現頻度（上位 10 種）

Cookie ヘッダはサーバから情報を送り込むことが可能なヘッダとして利用頻度が高く，正常なクライアントであれば正しく処理される可能性が高いと判断した．

さらに，代表的なブラウザの Cookie ヘッダ受付処理に関する標準設定を調査した．本来のサーバから発行された Cookie に見えるようにする必要があるかどうかも考慮するため，サードパーティ Cookie の設定も合わせて確認したところ，表 5.3 の通りであった．

ブラウザ	Cookie	サードパーティ Cookie
InternetExploler	有効	有効
Firefox	有効	有効
Safari	有効	無効
Chrome	有効	有効

表 5.3 Cookie 標準設定

意図的にユーザが設定を変更したり，CookieMonster[11]のようなアドオンを導入したりしなければ，Cookie は標準的に有効化されている．ただし，サードパーティ Cookie を無効とするブラウザが存在するため，プロキシサーバで発

行する Cookie ヘッダはファーストパーティとなるようドメイン設定を行う必要がある．ログインを伴う Web サイトなどは，Cookie ヘッダを無効にするとセッション管理ができないなど，閲覧に制限が出ることも多い．これらを理由として，一般の利用においてブラウザは標準的に Cookie ヘッダを処理すると考えてよいとした．

２）ETag ヘッダ

Cookie ヘッダと同種のヘッダとして，クライアントのキャッシュ制御に関連する ETag ヘッダの使用が可能かどうかについても調査を行った．この場合，サーバのレスポンスに ETag ヘッダを挿入し，次のアクセスで If-None-Match ヘッダが返ってくるかどうかを確認することとなる．しかし，ETag ヘッダは画像など静的コンテンツに付与されるタグであり，画像データなどが無ければヘッダを挿入できず効果が限定的となる．よって，本提案では検討を除外した．

5.5.1.2 コンテンツ挿入

JavaScript をコンテンツに挿入し，クライアントでスクリプトを動作させることによって応答を確認する検討を行った．その結果，JavaScript の挿入には以下のように複数の問題があることがわかった．

・コンテンツ圧縮

Content-Encoding ヘッダで zip が指定されている場合，圧縮されたコンテンツをプロキシサーバで一旦展開してからスクリプトを挿入する必要があり，円滑な通信を阻害する可能性がある．

・コンテンツの分割送信

Transfer-Encoding ヘッダで chunked が指定されている場合，分断されたコンテンツをプロキシサーバで一旦結合してからスクリプトを挿入する必要があり，円滑な通信を阻害する可能性がある．

・ヘッダとボディの不整合

HTTP プロトコルの仕様上，HTTP ヘッダで指定されたコンテンツの種類と HTTP ボディのコンテンツの種類の組み合わせが正しいことが保証されない．そのため，ヘッダの確認のみでコンテンツ変更を行うかどうか判断ができない．

このように，コンテンツに JavaScript のようなデータ挿入を行うことは容易ではないと考えられるため，本提案では検討を除外し今後の課題とした．

5.5.1.3 バックドアでの Cookie 使用率の調査

次に，バックドア通信ではどの程度 Cookie ヘッダが使用されているかを確認した．Malware Traffic Patterns[12]で公開されている 2013 年 6 月末日までの不正プログラム通信のうち，Cookie ヘッダを使用したものがどの程度存在するか調査を行った．結果，HTTP を使う 172 の不正プログラム通信のうち，158（Family 数 141）は Cookie ヘッダを含んでおらず，表 5.4 にあげる限られたバックドア（通信パターン 14，Family 数 10）のみ Cookie ヘッダが含まれていた．

以上より，一般的に利用されており，バックドアでは利用されにくい挿入可能な情報として Cookie ヘッダを選定し，検証を行うこととした．

Family	GET/POST pattern の一部
backdoor?	GET /18110123/page_32262 308. html
Banechant 1	GET /IGKKT
Beebus	GET /windosdate/v6/default.aspx?
Cookies Cookiebag Dalbot	GET /1799.asp
Cookies Cookiebag Dalbot	GET /3961.html
Cookies Cookiebag Dalbot	GET /8223.asp
Cookies Cookiebag Dalbot	GET /indexs.zip
Letsgo TabMsgSQL	GET /safe/1.asp?
Letsgo TabMsgSQL	GET /safe/1.asp?
Netravler	GET /fly/2013/2011/nettraveler.asp?
Netravler	GET /nt2011/zy/nettraveler.asp?
NTESSESS	GET /6K8gL8.html
Tarsip Eclipse	GET /blg7_8newtpl/image/7/7_12/images/redir?
WEBC2-Clover	GET /Default.asp
ZeroAccess / Sirefef	GET /stat2.php?

表 5.4 Cookie を使用したバックドア通信

5.5.2 提案手法概要

ここでは Cookie ヘッダを挿入することによりバックドアを検出する基本的な手法を概説する．

1）クライアントからの初期接続時

クライアントから HTTP のリクエストをプロキシサーバで受ける（この時の

URL を host/path とする)．プロキシサーバはレスポンスヘッダに Set-cookie ヘッダを加え，プロキシサーバ独自の値を持つ Cookie ヘッダ（以下 Cookie(P)と表記）を挿入する．これにより，クライアントはプロキシサーバが挿入した Cookie(P)の処理を行う．

２）クライアントからの再接続時

正規のブラウザであれば host/path への再リクエスト時に Cookie(P)を入れて送信するはずである．しかし，クライアントからのリクエストヘッダに入るはずの Cookie(P)がない場合は Cookie(P)の処理がされていないということであり，正規のブラウザではない可能性がある．プロキシサーバで Cookie(P)の発行状態を管理し，発行したはずの Cookie(P)を返さないリクエストをバックドア通信として識別することで動作の違いを明らかにすることが可能となる．

以上をまとめると図 5.5 のとおりとなる．図中の上段が初期接続時の動作，中段が正常なクライアント，下段がバックドアの動作となる．

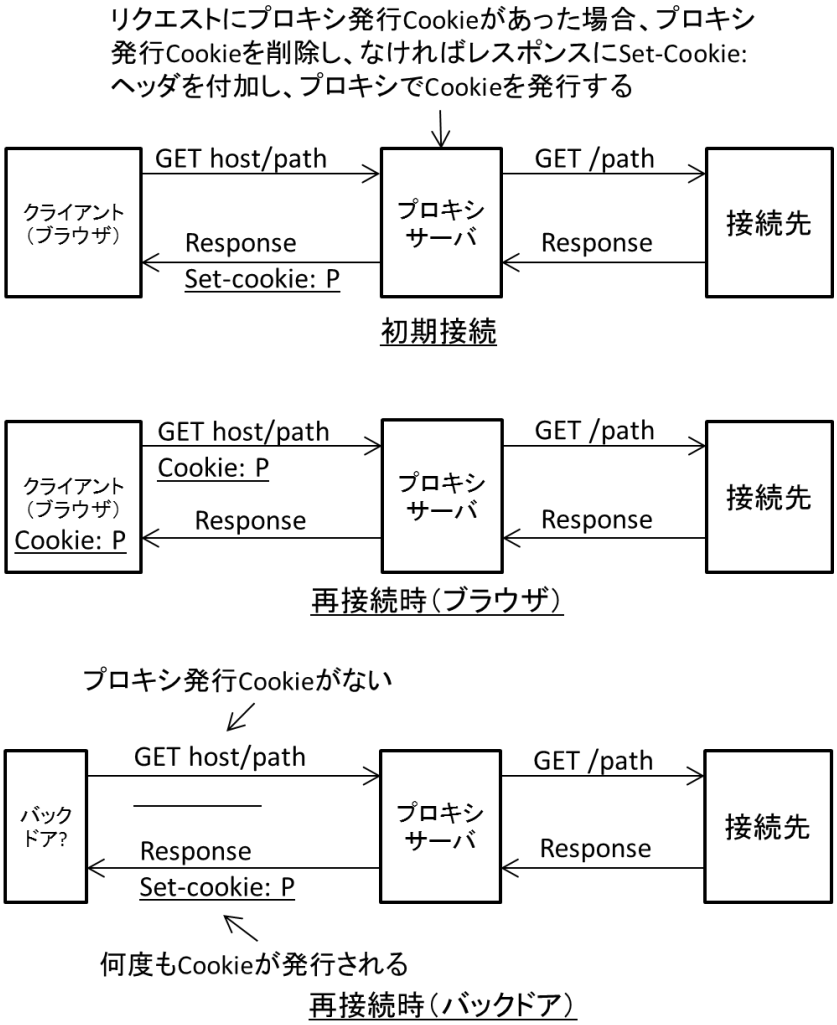

図 5.5 バックドア通信の検出

5.5.3 バックドア検出の流れ

提案手法では基本的に，プロキシサーバから Set-Cookie ヘッダを挿入したにもかかわらず，Cookie(P)がクライアントから送られてこない場合は処理が正しく行われていないと判断し，バックドアの可能性があるとした．しかし実際には，Cookie(P)の生存期間が終了して正規の処理が行われた結果クライアントから Cookie(P)が送られてこない場合，バックドアが偽物の Cookie ヘッダを送ってくる場合など，数多くの状態が考えられる．また，Cookie ヘッダは再リクエスト時にブラウザから送信されるが，ヘッダの挿入は初回レスポンス時に行われるため，プロキシサーバでそのトランザクションにおける Cookie の発行状態を正しく管理しなければならない．そこで Cookie ヘッダの発行状態として表 5.5 を定義し，リクエスト処理時にその状態を確認することで想定される誤検知を排除することとした．以下，図 5.6 にリクエスト処理時のバックドア検出フローを示すとともに，下記で説明を行う．

１）HTTP リクエストがあると，クライアントの IP アドレスと宛先サイトの組み合わせをキーとしてプロキシサーバに保存する．
２）プロキシサーバにキーが存在しない，かつクライアントからの通信に Cookie ヘッダそのものが存在しない場合，初回接続と考える．その場合の Cookie ヘッダの状態を(set)とする．
３）プロキシサーバにキーが存在しないがクライアントからの通信に Cookie ヘッダは存在し，かつ Cookie(P)は含まれていない場合，既存の Cookie ヘッダに Cookie(P)を付け加える．その場合の Cookie ヘッダの状態を(addvalue)とする．
４）プロキシサーバにキーが存在しないがクライアントからの通信に Cookie ヘッダは存在し，かつ Cookie(P)が含まれている場合は，管理情報の不整合と考える．その場合の Cookie ヘッダの状態を(inconsistency)とする．
５）プロキシサーバにキーが存在するがクライアントからの通信に Cookie ヘッダ自体が存在しない場合は，不正処理として Cookie ヘッダの状態を（suspicious）とする．
６）プロキシサーバにキーが存在かつクライアントからの通信に Cookie ヘッダが存在し，かつ Cookie(P)が含まれない場合は，管理情報の不整合と考える．その場合の Cookie ヘッダの状態を(inconsistency)とする．
７）プロキシサーバにキーが存在かつクライアントからの通信に Cookie ヘッダが存在し，かつ Cookie(P)が含まれているが値が正しくない場合は，管理情報の不整合と考える．その場合の Cookie ヘッダの状態を(resetvalue)とする．

８）プロキシサーバにキーが存在かつクライアントからの通信に Cookie ヘッダが存在し，かつ値の正しい Cookie(P)が含まれる場合は，正常と判断する．その場合の Cookie ヘッダの状態を(noproblem)とする．
９）最後に Cookie(P)が存在する場合は削除して Cookie ヘッダを元に戻し，宛先サーバに中継を行う．

Cookie(P)の不整合がおきる(4)(5)(6)(7)の状態は疑わしい通信として検出することが可能である．しかし，5.5.1 項のとおり Cookie ヘッダを使用するバックドアが限られていることから，本論文では Cookie が使用されない状態である，(5)suspicious をバックドアとして検出することとした．

set	Cookie(P)の挿入
addvalue	既存の Cookie ヘッダに Cookie(P)を追加
resetvalue	Cookie(P)をリセット
neverset	Cookie(P)を挿入しない
suspicious	Cookie(P)がない（バックドアとして検出）
inconsistency	Cookie(P)の値がデータベースの値と一致しない

表 5.5 Cookie ヘッダ状態の定義

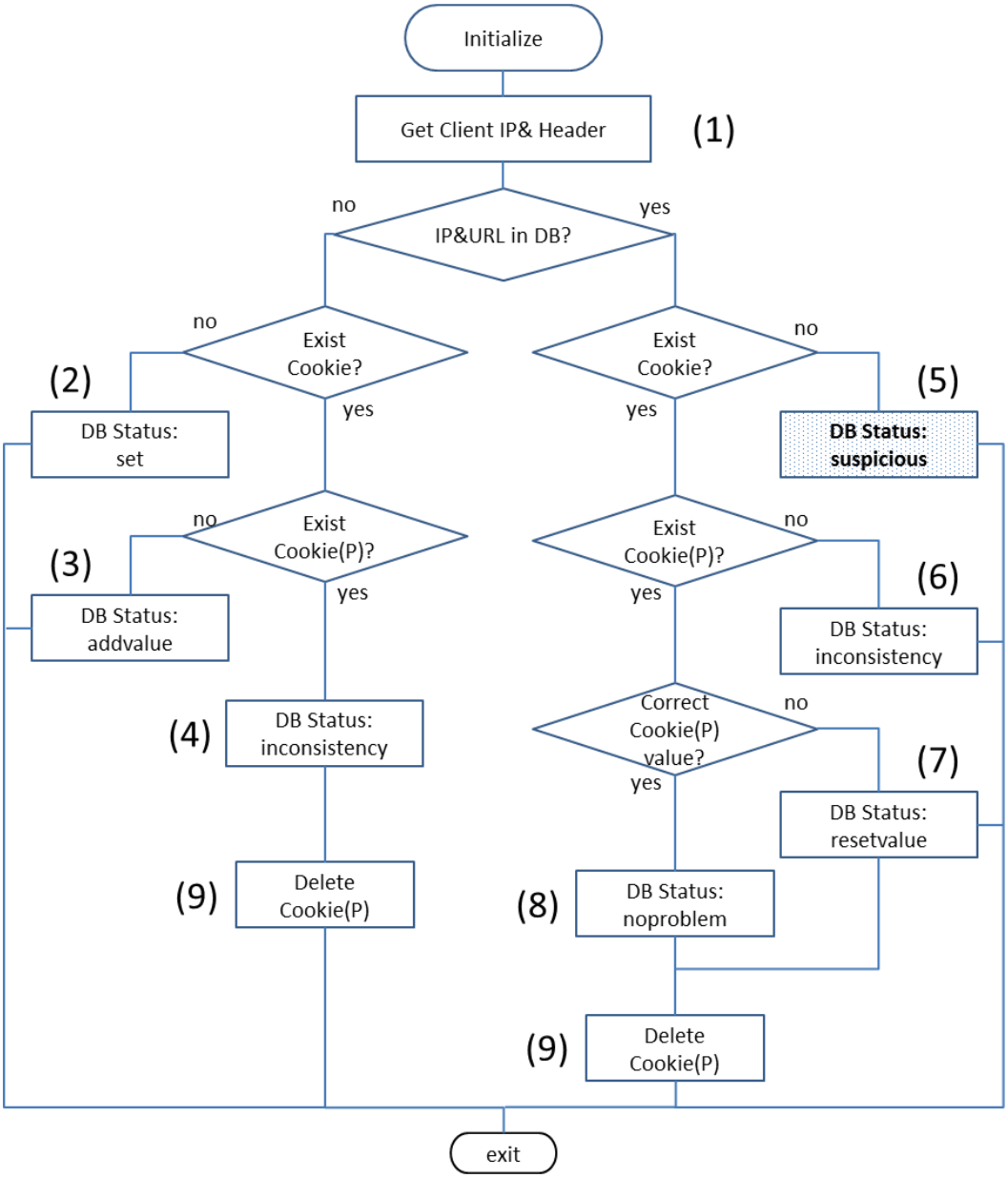

図 5.6 バックドア検出の流れ

5.6 実装

本提案手法はプロキシサーバでリアルタイムに HTTP ヘッダの追加，削除を行うことを想定している．また，前章で述べたようにどのクライアントからどの宛先サイトへのアクセスに Cookie(P)を挿入したといった，Cookie(P)の発行状態管理機能が必要となる．既存のオープンソースプロキシサーバである Squid[13]等の設定のみを使って実装を行うことは実用面を考慮した場合に有利であるが，状態管理の DB が必要なことや，ヘッダの操作に条件判断を伴うために，単純な設定のみでの実装は困難と考えられる．そこで今回は，Man In

The Middle 方式で HTTP 通信を操作することに特化したプロキシサーバである，ProxPy[14]を利用し，テスト実装を行うこととした．ProxPy は python で記述されており，プラグインプログラムを開発することによって，HTTP 中継時に追加処理を行うことが可能となる．なお，ProxPy のプラグインインターフェースは，クライアントの IP アドレスをプラグインプログラムに渡すことができない仕様であるため，今回は，IP アドレスの受け渡しが可能となるようにプラグインインターフェースを改修した．加えて，ヘッダを削除するルーチンが存在しなかったため，削除ルーチンの追加を行った．また，ヘッダとその値を追加するルーチンにおいて，元の通信に追加したいヘッダそのものが存在しない場合に，指定したヘッダも値も追加できないというバグが存在したため，その修正を行った．なお，プラグインのソースコードは 6KB，コメントを除いた行数は 117 行となった．Cookie(P)の発行状態管理，クライアント IP と宛先サイトの管理は，オープンソースのデータベースである，redis[15]を用いることとした．提案手法は扱う情報がシンプルであり，動作の高速な Key Value Store 型のデータベース利用が適当であると判断した．データベースはプロキシサーバで中継したクライアントの IP アドレスと宛先サイトの組み合わせを key とし， 表 5.5 の状態における Cookie(P)発行回数を value とした．その他，今回のテスト実装で使用した環境は表 5.6 のとおりである．

Function	Software	Version
OS	Ubuntu	12.04
Proxy	ProxPy	r27
Database	Redis	2.0.3
Language	Python	2.7

表 5.6 テスト実装で使用した環境

下記に初回アクセス時に挿入される Cookie の値をプロキシサーバで表示した例を図 5.7 に示す．"ProxySessionID=1"がプロキシサーバにより挿入された独自の値となる．その他に，3rd party Cookie とならないように host パラメータや，Cookie に持続性を持たせるための expires など Cookie ヘッダを形成するパラメータをあわせて Cookie(P)として生成している．

```
10.2.1.97:www.yahoo.co.jp ProxySessionID=1;
expires=Mon, 30-Jul-2014 23:59:59 GMT; path=/;
host=www.yahoo.co.jp
```

図 5.7 初期アクセス時のログ

5.7 提案手法の評価

実環境を用いて提案手法の効果を検証した．本章では評価条件，評価内容およびその結果を記す．

5.7.1 評価条件

1）評価に用いたシステム環境

評価に用いたシステム環境を図 5.8 に，機器スペックを表 5.7 に示す．プロキシサーバには Linux を，クライアントには Windows を使用し，ブラウザおよびバックドアを動作させた．

項目	サーバ	クライアント
CPU	Atom D510 1.66GHz	Athlon64x2 3GHz
Memory	2GB	8GB
DISK	40GB SSD	120GB SSD
OS	Ubuntu 12.04LTS	Windows7 Pro. SP1

表 5.7 評価用機器スペック

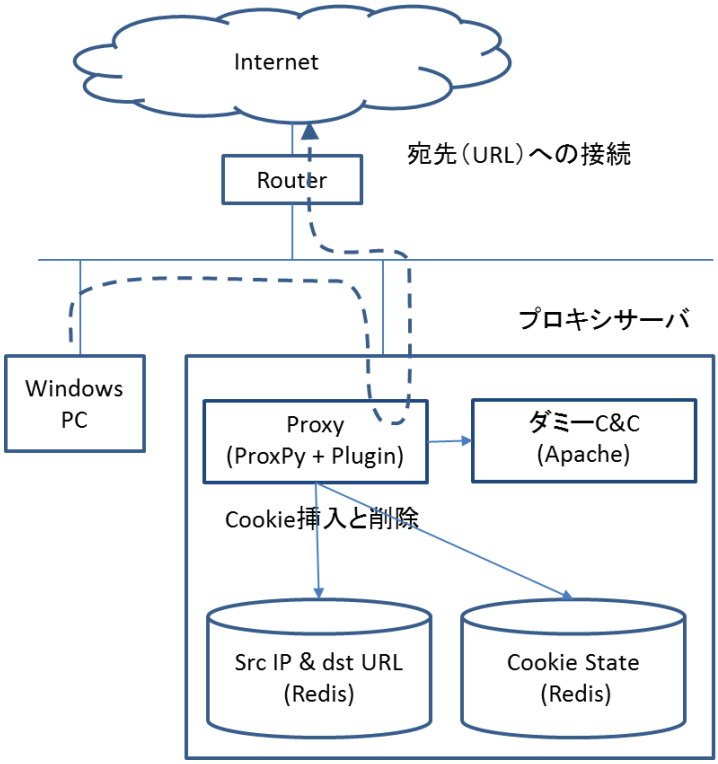

図 5.8 評価環境

２）使用したブラウザ

ブラウザは Firefox r24，および InternetExplorer 9 を使用し，プロキシサーバを経由して Web ブラウジングを行った．

３）使用したバックドア

標的型攻撃で使用されたバックドアプログラムは表 5.8 のものを選定した．選定理由を以下に記す．

標的型攻撃に利用されるバックドアは動作環境への依存性が高い．例えば，特定アプリケーションの脆弱性を利用して侵入し，動作するバックドアは，そのアプリケーションが存在しなければ動作しない．また，C&C サーバの疎通確認で失敗した場合に動作を停止するようなバックドアは， C&C サーバが動作していなければバックドア自体も活動しない．しかし，実際に活動している C&C サーバに接続することは危険であるし，また，検体が入手可能な時点で C&C サーバが動作していないこともあるため，その場合もテストができない．

これらの理由から，評価用バックドアプログラムは次の条件に適合するものを選定した．

・標的型攻撃で使用されていること．
・HTTP で通信を行うこと．
・プロキシサーバを使用すること．
・バックドア検体が手に入ること．
・今回用意した評価環境で動作すること．
・ダミーの C&C サーバがあれば動作すること

実際の動作においては，安全性を考慮し，動作中の C&C サーバに通信しないように，ダミーの C&C サーバを用意した．さらにメールの添付ファイルを開いた結果設置されるバックドアプログラムを抽出し，評価環境で動作するように設定を行った．具体的には，クライアントのプロキシサーバ設定を試験環境にあわせ，C&C サーバは内部ネットワークに用意した Web サーバを示すよう名前解決情報を差し替え，その Web サーバで C&C サーバが動作しているように見せかけるため偽の応答を返す．この方法により，本物の C&C サーバへアクセスしないようにした．

名称	ベンダ
BKDR_MALEX.RG	Trend Micro
BKDR_DEMTRANC.R	Trend Micro
Trojan.Win32.Zapchast.pbs	Kaspersky

表 5.8 バックドアサンプル

5.7.2 評価項目

評価内容について，以下の項目に関してテストを行った.

１）正常な通信の誤検出

まず，正常なブラウザからの通信を不正なものとして検出する誤検出の可能性を，ブラウザで Web サイトの閲覧を行い評価した．閲覧する Web サイトには Web サイトのアクセス順位を公開している Alexa[16]によりランキングされた世界の上位 100 サイトで HTTP 接続となるものを使用した.

テストに使用したドメインの一部を表 5.9 に記す.

２）バックドアの検出可否

バックドアを検出するかを，表 5.8 で用意したバックドアを実際に動作させることにより評価を行った.

google.com
facebook.com
youtube.com
yahoo.com
baidu.com
wikipedia.org
qq.com
linkedin.com
live.com
twitter.com

表 5.9 アクセスしたドメイン（一部）

３）処理速度への影響

Cookie(P)の挿入における通信遅延の検証を行った．確認方法としては，ProxPy 単独で動作させた場合と，ProxPy でプラグインを動作させた場合それぞれにおいて，ブラウザを使用して特定サイトにアクセスした．ブラウザは

Firefox r24 を使用し，アクセス先のサイトは www.yahoo.co.jp のトップページとした．プラグイン動作単体の速度差分を確認するため，プロキシサーバのデータベースは初期化した状態で，ブラウザキャッシュを無効化したクライアント 1 台を使用して計測した．アクセス速度の計測は Firefox のアドオンプログラムである Firebug[17]を使用し，10 回のアクセスを 1 セットとし，それを 3 セット繰り返した．

5.7.3 評価結果

以下にそれぞれのテストにおける評価結果を示す．

１）正常な通信の誤検出

結果，テストに使用したサイトへの閲覧に対する誤検出（False positive）は発生しなかった．Cookie(P)がブラウザで正常に処理された場合の画面例を図 5.9 に示す．

図 5.9 通常のブラウザの振る舞い

２）バックドアの検出可否

3 つのバックドアのうち，BKDR_MALEX.RG のバックドア通信をリアルタイムで検出した．

```
10.2.1.79:www.aol.com : Cookie does not accept
10.2.1.79 : infected? Other access is  1 IP
{REQ #70} method: GET ; host: ('www.aol.com', 80) ;
path: /n/klut7NiFR10gVMcbNBNLqa ; proto: HTTP/1.1 ;
```

```
len(body): 0
  Accept-Language: ja
  Accept-Encoding: gzip, deflate
  Accept: image/gif, image/x-xbitmap, image/jpeg,
  image/pjpeg, application/x-shockwave-flash,
  application/x-ms-application, application/x-ms-xbap,
  application/vnd.ms-xpsdocument, application/xaml+xml,
  */*
  User-Agent: Mozilla/4.0 (compatible; MSIE 7.0;
  Windows NT 5.1; .NET CLR 2.0.50727; .NET CLR
  3.0.4506.2152; .NET CLR 3.5.30729)
  Host: www.aol.com
  UA-CPU: x86
  Proxy-Connection: Keep-Alive
}
```

上記は検出時のログ出力である．バックドアが疎通確認のために AOL へ接続しようとする通信に Set-Cookie を挿入し，次の接続で Cookie(P)が存在しないことを "Cookie is not accepted" というメッセージとともに出力している．一方，BKDR_DEMTRANC.R および Trojan.Win32.Zapchast.pbs は検出されなかった．理由は後述する．

３）処理速度への影響

結果は表 5.10 の通りとなった．今回開発したプラグインが動作していない場合と動作している場合を比較して，プラグインを動作させた場合はそれぞれ 1.08 倍，1.04 倍，1.06 倍となり，平均すると約 6%の処理時間増加となった．

セット	プラグイン	平均アクセス時間（ｓ）
1	なし	16.16
1	あり	17.47
2	なし	17.81
2	あり	18.51
3	なし	16.66
3	あり	17.63

表 5.10 平均アクセス時間

5.8 考察

以下に検証の結果をもとにした考察を示すとともに，提案手法の有効性について述べる．

5.8.1 正常な通信の誤検出

本実装では正規の通信を誤検出することはなかった．5.5.3 項で述べたように，データベース上の管理情報とブラウザが保存している Cookie(P)との不整合が発生する可能性はある．その場合には Cookie をリセットし，再度挿入を行うことにより，正常なブラウザであれば次に応答があるため，誤検出を低減できていると考えられる．また，ブラウザではないがソフトウェアアップデートなどで通信を行うプログラムについて，HTTPS を利用するものは本論文の対象としていないが，HTTP を使う場合は誤検出を起こす可能性が否定できない．しかし，そのようなプログラムは接続先が固定的であり，複数のユーザから同じドメインへ接続されるため，ホワイトリストを併用することで誤検出を回避することが可能と考えられる．

5.8.2 バックドアの検出可否

提案方式はクライアントの実装の差異の確認であり，宛先 URL に依存しないバックドアの検出が可能であるため，バックドアから正規サイトへの疎通確認や，宛先が汎用的なホスティングでブラックリストが作れない場合，また，正規のサイトが改ざんされているような場合においても検出することが可能である．一方，挿入するヘッダとして Cookie を使用した本実装では，以下の場合においてバックドアを検出しない．以下でその考察を行う．

・HTTPOpenRequest API の使用

HTTPOpenRequest API[18]は InternetExplorer などが使用する API であるが，バックドアもこの API を使用することが可能である．5.5.1 項において，一般的なブラウザでは問題なく処理されるヘッダやデータのうち，バックドアプログラムではうまく処理されないものが存在すると仮定したが，この API は HTTP プロトコルのヘッダ処理を自動で行うため，バックドアがこの API を使用することで，Cookie ヘッダ処理を InternetExplorer と同等に行うことが可能となる．この場合，本実装での検出は困難となる．このような，OS が提供する高レベル API を使用せず，Winsock などの低レベル API を使用したバックドアや，固定的な Cookie ヘッダを出力するプログラムであれば検出可能と考えられる．また，Amir Houmansadr らの指摘によると，偽装された通信は正規の通信と何

らか異なる部分が存在すると考えられる[19]．正規の通信を完全に再現することは実装コストが高くなるため，ヘッダの出現順序の確認や，Cookie 以外のヘッダを組み合わせてデータを挿入するなどすることで，提案手法が有効な検出方法となる可能性がある．

・CONNECT メソッドの使用

CONNECT メソッドは主に SSL 通信で使用されるメソッドであり，標準的にポート番号 443 を使用して通信内容を暗号化する．本方式の対象は HTTP 通信を行うバックドアを前提としており，暗号化通信は対象としていないが，CONNECT メソッドそのものは TCP でトンネル接続をするための HTTP のメソッドの一つである．暗号化に限って使用されるとは限らないため，今後の議論として記載する．

まず，CONNECT メソッドを使用して暗号化通信が行われた場合，プロキシサーバで復号することにより通常の HTTP となるため，ヘッダを挿入することは可能である．暗号化通信をプロキシサーバで復号するためには，宛先サイトとプロキシサーバ，プロキシサーバとクライアントの間で異なる鍵を使用する手法が用いられるが，クライアントから見て本来のサイトと異なる鍵が使用されることになる．利用に支障をきたさないためにはプロキシサーバの鍵をクライアントに事前配布しておくといった対処が必要となるため，運用には注意を要する．また，CONNECT メソッドが暗号化通信として使用されていない場合は，プロキシサーバでヘッダを挿入することが可能である．しかし，どのようなヘッダを挿入するか，また，CONNECT メソッドがブラウザでどのように処理されるかは今後さらなる検証が必要である．

・ユーザの意図的な設定による Cookie の無効化

この場合，特定の利用者でアラートが多発すると考えられ，意図的な無効化が容易に認知できると考えられる．クライアントの IP アドレスをホワイトリスト化するなどの対応を行う必要性がある．

5.8.3 処理速度への影響

プラグインを動作させた場合に，平均 6%の処理時間増が見られたが，検証用の実装は ProxPy およびプラグインプログラムとも高速性を追求した実装ではなく，分岐処理などで時間を要していると考えられる．しかし，ヘッダの挿入量が少なく（ドメインの長さにもよるが 100Byte 前後），データベースも単純なことから，遅延は実装上の課題と考えられ，プログラムのチューニングにより処理時間は圧縮できると考えられる．

5.8.4 適用範囲

提案手法の効果を高めるためには，運用環境について考慮する必要があると考えられる．まず，ユーザがプロキシサーバを利用することを前提としているため，原則として組織内ネットワークへの適用が前提となる．また，検出効果を高めるためには，5.4 節で挙げた，ネットワーク構成による対策などとの併用が望ましい．加えて，検出される情報としては送信元の IP アドレスと宛先 URL の組み合わせであるため，標的型攻撃を受けているユーザを特定するなどの対策につなげるには，送信元の IP アドレスを誰が利用しているかなどの情報が管理されている必要がある．DHCP などでユーザ端末のアドレスを動的に割り振るような場合は，DHCP リース情報などの管理が行われていなければならない．すでにプロキシサーバを運用している組織内で本提案手法を適用する場合には，ユーザ端末に対して設定追加やソフトウェアインストールなどを要求する必要がなく，導入への負荷は低いと思われる．また，クライアントとプロキシサーバ間でのみ通信が行われるため，宛先 URL のブラックリストなど他のネットワークセキュリティ対策との併用も容易であると考えられる．

5.8.5 提案手法の利点

以上より，関連研究で挙げた既存の対策などと比較して本提案手法の利点と考えられることを以下にまとめる．

・リアルタイムで正規のブラウザではない通信を検出することが可能．
・通信先が正常なサイトであってもバックドアを検出することが可能．
・プロキシサーバで実装するため，クライアントに新たなソフトウェアを導入する必要がない．
・クライアントの応答を確認しているため，パターンが存在しないバックドアも検出することが可能．
・接続先 URL のブラックリストのような，他のフィルタ方式と併用が容易．
・特別な装置を必要とせず．プロキシサーバとして導入が可能．
・処理速度への影響が少ない．
・統計値を使用しないため学習が必要なく，統計誤差による誤検出はない．

5.9 まとめ

攻撃者は標的型メール等を使い，内部ネットワークに侵入し，遠隔操作用バックドアを動作させ，情報窃取を行う．中でも HTTP プロキシサーバを使って遠隔操作を行うバックドアはユーザが通常行う通信と区別が困難なため，検出

されないか，検出までに時間を要する．そこで本章では，プロキシサーバで通信を中継する際に元の通信には無い情報を挿入し，クライアントの応答を確認することでリアルタイムにバックドア通信を検出する手法を提案した．挿入する情報の選定に当たり，一般利用者のデータおよび既知の不正プログラム通信の情報を調査し，挿入する情報として Cookie を使用することとした．誤検出を排除するための条件を考慮し，実装を行い，提案手法の検証を行った．検証は通常の利用をバックドアとして誤検出しないかどうか，バックドアを検出するか，速度的に影響がないかの観点で実施した．結果，一定の性能を確保しつつ，バックドア通信をリアルタイムで検出することに成功した．一方，提案手法では検出が困難なバックドアの存在も確認された．これらの対策については今後の課題として研究していきたいと考えている．

参考文献

［１］情報処理推進機構：「標的型メール攻撃」の対策に向けたシステム設計ガイド，情報処理推進機構（オンライン），入手先 <https://www.ipa.go.jp/files/000033897.pdf>（参照 2014-12-11）．
［２］Sood, A. and Enbody, R.：Targeted Cyberattacks: A Superset of Advanced Persistent Threats, Security Privacy, IEEE, Vol. 11, No. 1, pp. 54-61 (2013).
［３］情報処理推進機構：『高度標的型攻撃』対策に向けたシステム設計ガイド，情報処理推進機構（オンライン），入手先 <https://www.ipa.go.jp/files/000042039.pdf>（参照 2014-12-11）
［４］Schwenk, G. and Rieck, K.：Adaptive Detection of Covert Communication in HTTP Requests, Computer Network Defense (EC2ND), 2011 Seventh European Conference on, pp. 25-32 (2011).
［５］jun Ding, Y. and dong Cai, W.：A method for HTTP-tunnel detection based on statistical features of traffic, Communication Software and Networks (ICCSN),2011 IEEE 3rd International Conference on, pp. 247-250 (2011).
［６］Giura, P. and Wang, W.：A Context-Based Detection Framework for Advanced Persistent Threats, Cyber Security (CyberSecurity), 2012 International Conference on, pp. 69-74 (2012).
［７］Balduzzi, M., Ciangaglini, V. and McArdle, R.：Targeted attacks detection with SPuNge, Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on, pp. 185-194 (2013).
［８］de Vries, J., Hoogstraaten, H., van den Berg, J. and Daskapan, S.：Systems for Detecting Advanced Persistent Threats: A Development Roadmap Using Intelligent Data Analysis, Cyber Security (CyberSecurity), 2012 International Conference on, pp. 54-61 (2012).
［９］PaloAltoNetworks: Next-Generation Firewalls, Palo Alto Networks Inc. (online), available from <https://www.paloaltonetworks.com/> (accessed 2014-12-11).
［１０］FireEye: FireEye, FireEye Inc. (online), available from <https://www.fireeye.com/> (accessed 2014-12-11).
［１１］s1tony: Cookie Monster, (online), available from <https://addons.mozilla.org/ja/firefox/addon/cookie-monster/> (accessed 2014-12-11).
［１２］Mila: Library of Malware Traffic Patterns, Deep End REsearch (online), available from <http://www.deependresearch.org/2013/04/library-of-malware-traffic-patterns.html> (accessed 2014-12-11).
［１３］SquidProject: Squid: Optimising Web Delivery, The Squid project (online), available from <http://www.squid-cache.org/> (accessed 2014-12-11).

[１４] roberto.paleari@gmail.com and alessandro.reina@gmail.com: proxpy A Python HTTP/HTTPS Proxy, (online), available from <https://code.google.com/p/proxpy/> (accessed 2014-12-11).
[１５] San filippo, S.: Redis, Pivotal Software, Inc. (online), available from <http://redis.io/> (accessed 2014-12-11).
[１６] AlexaInternet: Alexa Top 500 Global Sites, Alexa Internet Inc. (online), available from <http://www.alexa.com/topsites> (accessed 2014-12-11).
[１７] Hewitt, J.: Firebug, FirebugWorkingGroup (online), available from <https://addons.mozilla.org/ja/firefox/addon/firebug/> (accessed 2014-12-11).
[１８] Microsoft: HttpOpenRequest function, Microsoft Corporation (online), available from <http://msdn.microsoft.com/en-us/library/windows/desktop/aa384233%28v=vs.85%29.aspx> (accessed 2014-12-11).
[１９] Houmansadr, A., Brubaker, C. and Shmatikov, V.: The Parrot Is Dead: Observing Unobservable Network Communications, Security and Privacy (SP), 2013 IEEE Symposium on, pp. 65-79 (2013).

第6章 結論

近年インターネット環境は重要な情報をやり取りするだけの信頼性を持った社会インフラとなった．規模は拡大の一途をたどり，インターネットを支える技術やシステム構成はすでに把握しきれないほどに複雑化している．しかし，ネットワークシステムはいまだにインターネット初期から変わらない手法と技術で作られ，そして守られている．標的型攻撃はそのような従来型の設計・構築方法の限界をまざまざと見せつける，進化した攻撃である．このような攻撃に対抗するためには，防御側も従来型の発想を捨て，新しい方法に取り組んでいかなければならず，本論文ではそのために新たな取り組みを行った．

本論文での取り組みについて以下にまとめる．

第 1 章では，標的型攻撃が登場する背景を情報の質の変化，システムの複雑さの変化，攻撃手法の変化の観点から明らかにし，標的型攻撃手法を概説した．

第 2 章では，本研究で対象とするネットワークシステムがどのようなものであるか，それらに対してどのような攻撃が行われているかを明らかにし，現状の対策の限界を示した．それらを考慮して，新たな標的型攻撃対策の方向性を示した．

第 3 章では，標的型攻撃を考慮した，新たなネットワークシステムの設計手法を提案した．ネットワークシステムのレイヤ構造に着目し，レイヤ内のアクセス情報とレイヤ間の依存関係を明らかにすることで，ネットワークシステム設計における定量評価が可能となることを示し，その応用として攻撃による影響を測る尺度となる脆弱性影響度を提案した．

第 4 章では，段階的に行われる攻撃を想定して，攻撃活動のネットワーク動作をモデリングし，提案ネットワークモデルと合わせることによって，ネットワークシステム上でどのように攻撃が進行するかといったことがシミュレーションできることを示した．

第 5 章では，標的型攻撃の初期段階から使用されるバックドア通信に着目し，プロキシサーバを使ってダミーデータをクライアントに送り込み，その応答を見ることで不正プログラムによる通信かどうかを識別する手法を提案した．その提案手法を実装し，実際に標的型攻撃で使用された不正プログラムを動作させテストを行ったところ，攻撃の初期段階で不正プログラムによる偽装通信を検出することが可能であることが確認された．これにより，早期の攻撃予兆検出が可能であることを示した．

総合すると本研究では，マルチレイヤ・マルチファンクションのモデルを用いた，シミュレーションによる定量的評価可能な，ネットワークシステムのセ

キュリティ設計手法，および検証手法の基礎技術確立，また，中継装置を利用した情報挿入による不正プログラムの検出手法の基礎技術確立を行った．その結果，ネットワークシステム全体で行う多段階防御による標的型攻撃への新たな対策手法の基礎技術確立という成果が得られた（図 6.1）．

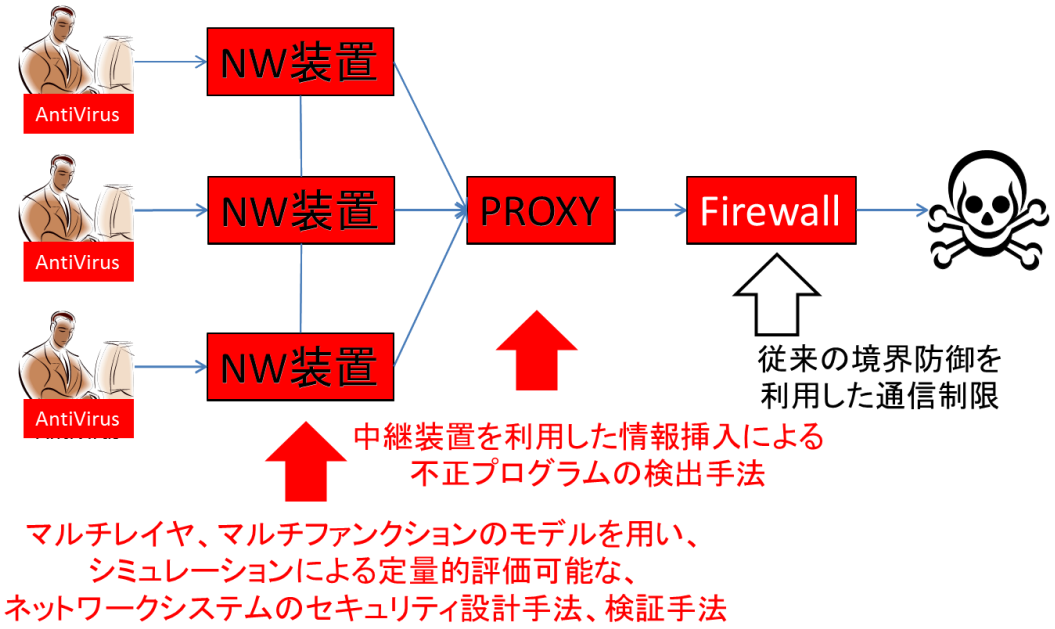

図 6.1 本研究の成果

これまでに述べたとおり，多層構造で多くの機能を同時に提供するような特徴を持ったネットワークシステムにおける，セキュリティ設計の標準的手法が存在しない．そのことが問題の根底にあると考えられる．設計内容，構築機器の機能，通信量，利用者数など，インターネットをとりまく環境における，ありとあらゆるものの規模がすでに人のスキルだけでは処理できないほどに巨大化しているにもかかわらず，ネットワークシステムは旧来の設計手法，防御手法を使い続けている．標的型攻撃は従来の設計手法・防御手法からの脱却を促しているとも言えるだろう．ネットワークシステムの設計・構築・運用も含めて，従来とは違う視点で新たな方法を導入しなければ対策は困難であり，人手でできる範囲で対処が可能な状況ではなくなってきているが，本論文で提案したアクセス制御設計手法やデータモデル，また，その考え方を応用することによって，標的型攻撃の迅速な検出や対処を自動で行うということも今後可能となるであろうと考えられる．

近年実用化されたクラウドコンピューティング環境においても，セキュリティ評価可能なモデルを使用して設計することが期待される．画面上で行うシス

テム設計から，安全性が計算によって導出され，その設計内容に沿ってシステム構築が自動的に行われるようになることで，より安全なクラウドシステムが利用可能となる．本論文で提案したモデルは標的型攻撃に限らず，様々なセキュリティ事象に対するネットワークシステム設計を積極的にコンピュータで支援することに対しても適用が可能であるため，今後様々な応用が期待される．

本論文では標的型攻撃対策の基盤技術として新たなネットワークシステムの設計手法や実装方法のコンセプトを提案した．一方，数百台，数千台規模のネットワークシステムへの適用を考慮した設計・構築・運用自動化，さらに現在稼働中のネットワークシステムへの研究成果の適用等を考慮した場合には，今後解決が必要な複数の課題が存在している．具体的には，さらなる研究項目として，次のような内容が今後の検討課題として考えられる．

- 提案ネットワークモデルをベースとした，より詳細なデータモデル定義の研究
- 実稼働環境から収集可能な情報を利用し，ネットワークモデルへと変換を行う手法の研究
- 標的型攻撃シミュレーションにおける，攻撃モデルの詳細化およびシミュレーション精度向上に関する研究
- CVSS 等の公開情報とデータモデルを連携する技術の研究
- 既存の攻撃検知機器や攻撃検知手法から得られる情報と，設計情報やネットワークモデルの自動照合による攻撃検出技術に関する研究
- 中継装置での情報挿入による不正プログラムの検出精度向上に関する研究
- 中継装置での情報挿入による不正プログラム検出と他の検出手法との組み合わせによる攻撃検出および防御性能向上に関する研究
- 攻撃検知機器およびセキュリティ対策機器間の自動的な防御連携技術に関する研究

本研究の対象は実際の企業や組織で利用されているネットワークシステムであり，実証実験は困難なことが多い．しかし，現在はクラウドコンピューティングが実用的なソリューションとなっており，本論文で提案した設計手法やデータモデル，対策手法を実運用環境に載せるための実証実験も行いやすくなっている．今後本論文の成果をベースとしてさらなる研究を進め，成果を活用していくことで，抜本的かつ実用的な標的型攻撃対策が行われることを目指していきたいと考えている．

第7章 発表論文

【本論文で使用した論文】

[1] 加藤雅彦，小出洋，金岡晃，松川博英，前田典彦，岡本栄司，HTTP プロキシサーバでの Cookie 挿入によるバックドア通信の検出，情報処理学会論文誌 Vol. 55 No. 9 pp. 2008-2020, 2014

[2] Masahiko Kato, Takumi Matsunami, Akira Kanaoka, Hiroshi Koide, Eiji Okamoto, “Tracing Advanced Persistent Threats in Networked Systems”, Automated Security Management, pp 179-187, 2013

[3] 金岡晃，原田敏樹，加藤雅彦，勝野恭治，岡本栄司，安全なネットワークシステム設計のためのマルチレイヤネットワークモデルの提案と応用，情報処理学会論文誌 Vol. 51, No. 9, pp. 1726-1735, 2010.

【その他の誌上発表論文】

[1] Ichita Higurashi, Akira Kanaoka, Masahiko Kato, Eiji Okamoto, “Detection of Unexpected Services and Communication Paths in Networked Systems”, Journal of Information Processing, Vol. 21, No. 4, pp. 632-639, 2013.

[2] 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 年 12 月.

[3] 原田敏樹，金岡晃，加藤雅彦，勝野恭治，岡本栄司，ネットワークシステムにおける脆弱性影響の測定手法とシステム実装，情報処理学会論文誌 Vol. 52 No. 9 pp. 2613-2623, 2011.

[4] 新 麗，二宮 恵，加藤 雅彦，″マルチドメイン環境におけるネットワークシステム管理制御機構の設計″，電子情報通信学会技術報告[インターネットアーキテクチャ]，pp.71--76，IA2008-60，2009 年 1 月.

[5] 二宮 恵，新 麗，加藤 雅彦，″データモデルに基づくネットワークシステム管理のプロトタイプ構築″，電子情報通信学会技術報告[インターネットアーキテクチャ]，pp.31--36，IA2008-46，2008 年 11 月.

[6] 新 麗，二宮 恵，加藤 雅彦，″ネットワークシステム管理のための構成情報データモデルの設計″，電子情報通信学会技術報告[インターネットアーキテクチャ],pp.25--30，IA2008-45，2008 年 11 月.

【その他の口頭発表】

[1] 加藤 雅彦（株式会社インターネットイニシアティブ/筑波大学），小出 洋

(九州工業大学), 金岡 晃 (東邦大学), 松川 博英 (トレンドマイクロ株式会社), 前田 典彦 (株式会社カスペルスキー), 岡本 栄司 (筑波大学), "HTTP Proxy を使った Cookie 挿入による不正通信の検出", CSS2013
[2] 陳 帥 (筑波大学), 金岡 晃 (情報通信研究機構), 松尾 真一郎 (情報通信研究機構), 加藤 雅彦 (株式会社インターネットイニシアティブ), 須賀 祐治 (株式会社インターネットイニシアティブ), 岡本 栄司 (筑波大学), "モバイル端末のリスク分析と対策の自動適用手法", CSS2013
[3] 金岡 晃 (東邦大学), 加藤 雅彦 (筑波大学), 小出 洋 (九州工業大学), 岡本 栄司 (筑波大学), "組織内ネットワークとマルウェアのモデル化データを用いたマルウェア被害分析", CSS2013
[4] M. Kato, T. Matsunami, A. Kanaoka, H. Koide, and E. Okamoto, Tracing Attacks on Advanced Persistent Threat in Networked Systems (short paper), 5th Symposium on Configuration Analytics and Automation (SafeConfig 2012), 2012
[5] I. Higurashi, A. Kanaoka, M. Kato, and E. Okamoto, Discovery of Unexpected Services and Communication Paths in Networked Systems (short paper), 5th Symposium on Configuration Analytics and Automation (SafeConfig 2012), 2012
[6] M. Kato, T. Matsunami, A. Kanaoka, H. Koide and E. Okamoto: Tracing Attacks on Advanced Persistent Threat in Networked Systems, 21st USENIX Security Symposium Poster Session (Aug 2012)
[7] 三上 烈史, 吉田祥真, 小林良太郎(豊橋技科大), 加藤雅彦, 金岡晃(筑波大), 一時的アクセス権を用いたインターネット予約システムにおけるクライアントのスケジューリング, 第 11 回情報科学技術フォーラム(FIT2012), pp.229-232, 2012 年 9 月 4-6 日. (於 東京都小金井市 法政大学小金井キャンパス)
[8] 吉田祥真, 三上 烈史, 小林良太郎(豊橋技科大), 加藤雅彦, 金岡晃(筑波大), 複数台のおとりマシンによる HTTP-GET Flood 攻撃対策, 第 11 回情報科学技術フォーラム(FIT2012), pp.207-210, 2012 年 9 月 4-6 日. (於 東京都小金井市 法政大学小金井キャンパス)
[9] 日暮一太, 金岡晃, 加藤雅彦, 岡本栄司, 設計者の意図しないサービスと通信経路の発見手法, 情報処理学会 CSS2012, 2012
[10] 日暮一太, 金岡　晃, 加藤雅彦, 岡本栄司, マルチレイヤのネットワークトポロジ抽出手法, 第 10 回情報科学技技術フォーラム, 2011 年 9 月
[11] T. Harada, A. Kanaoka, E. Okamoto, M. Kato, “Identifying Potentially-Impacted Area using CVSS for Networked Systems”,

Proceedings of The First Workshop on Convergence Security and Privacy (CSnP), July. 2010
[12] 加藤雅彦, “IaaS システムのセキュリティ境界に関する一考察”, 第 39 回画像電子学会年次大会, June 2010.
[13] 金岡晃, 加藤雅彦, 岡本栄司, Web 感染型マルウェアリスク評価を可能とするネットワークトポロジ分析, 電子情報通信学会 ICSS 研究会, 2010.
[14] 金岡 晃, 原田 敏樹, 加藤 雅彦, 岡本 栄司：向きを持つマルチレイヤネットワークモデルの提案とセキュリティへの応用, 情報処理学会 コンピュータセキュリティ研究会, 2009 年 12 月
[15] 原田 敏樹, 金岡 晃, 岡本 栄司, 加藤 雅彦：ネットワークシステムにおける CVSS を用いた脆弱性影響範囲特定手法の検討, 電子情報通信学会 情報通信システムセキュリティ研究会, 2009 年 11 月
[16] 原田 敏樹, 金岡 晃, 岡本 栄司, 加藤 雅彦：CVSS を用いたネットワークシステムの危険度測定手法の検討, 電子情報通信学会 情報通信システムセキュリティ研究会, 2009 年 7 月
[17] A. Kanaoka, M. Katoh, N. Toudou, E. Okamoto, “Extraction of Parameters from Well Managed Networked System in Access Control”, Proceedings of The Fourth International Conference on Internet Monitoring and Protection (ICIMP2009), pp.56-61, May. 2009
[18] 原田 敏樹, 金岡 晃, 加藤 雅彦, 岡本 栄司：脆弱性情報提供 Web API “AVIP” の開発, 2009 年 暗号と情報セキュリティシンポジウム (SCIS2009), 2009 年 1 月
[19] 二宮 恵, 新 麗, 加藤 雅彦, 松尾 広大, 亀崎 真弓, 大野 邦夫, ″ネットワーク機器連携による情報家電利用シナリオの設計″, 画像電子学会第 22 回 VMA 研究会, 2009 年 1 月.
[20] 新 麗, 二宮 恵, 加藤 雅彦, 宮川 祥子, ″保育支援をめざした画像共有システムプロトタイプの設計″, 画像電子学会第 22 回 VMA 研究会, 2009 年 1 月.
[21] A. Kanaoka, M. Katoh, N. Toudou, E. Okamoto, “Networked System Modeling and its Access Control Characteristic Analysis”, Proceedings of World Academy of Science, Engineering and Technology (WASET), Vol. 35, pp.125-133, Nov. 2008
[22] 藤堂 伸勝, 加藤 雅彦, 金岡 晃, 岡本 栄司：ネットワークシステムにおける可用性測定の考察,コンピュータセキュリティシンポジウム 2008 (CSS2008), 2008 年 10 月
[23] 金岡 晃, 藤堂 伸勝, 加藤 雅彦, 岡本 栄司：適切なアクセス制御状態にあるネットワークシステムの特徴抽出,コンピュータセキュリティシンポジ

ウム 2008 (CSS2008), 2008 年 10 月
[24] 加藤 雅彦, 金岡 晃, 藤堂 伸勝, 岡本 栄司：ネットワークシステムにおける脆弱性影響度の定量化と可視化 ,コンピュータセキュリティシンポジウム 2008 (CSS2008), 2008 年 10 月
[25] 金岡 晃, 加藤 雅彦, 藤堂 伸勝, 岡本 栄司：アクセス制御の違いによる ネットワークシステムの特性変化に関する考察, 電子情報通信学会 情報通信システムセキュリティ研究会, 2008 年 9 月
[26] 新 麗, 二宮 恵, 加藤 雅彦, ″構成情報の仮想モデルによるネットワーク管理プロトタイプの構築″, 画像電子学会第 21 回 VMA 研究会, 2008 年 7 月.
[27] 新 麗, 二宮 恵, 加藤 雅彦, ″ユーザ主体のネットワーク利用を実現するネットワーク管理制御機構の提案″, 第 36 回画像電子学会年次大会, July 2008.
[28] 金岡 晃, 藤堂 伸勝, 加藤 雅彦, 岡本 栄司, “ネットワークシステムの安全性定量化に向けた新たな表現モデルとアクセス制御解析”, 2008 年 暗号と情報セキュリティシンポジウム (SCIS 2008)

謝辞

本論文を執筆するにあたり，貴重な時間を割いてご指導ご鞭撻をいただきました筑波大学大学院システム情報工学研究科リスク工学専攻岡本栄司教授に心から感謝いたします．また，本研究の遂行にあたり，数多くのご指導をいただきました東邦大学理学部情報科学科金岡晃講師，九州工業大学情報工学研究院知能情報工学研究系小出洋准教授に深く感謝いたします．お忙しい中学位審査の副主査をお引き受けいただいた，筑波大学大学院システム情報工学研究科リスク工学専攻片岸一起准教授，西出隆志准教授，金山直樹助教に深く感謝いたします．

本研究を遂行するために数々のご協力をいただいた，トレンドマイクロ株式会社松川博英様，株式会社カスペルスキー前田典彦様，九州工業大学小出研究室松浪拓海様はじめ研究室の皆様，日本 IBM 勝野恭治様，そして筑波大学岡本研究室原田敏樹様，藤堂伸勝様，勝倉辰之助様，日暮一太様に深く感謝いたします．また，多くの議論とともに示唆に富んだご指摘をいただいた，富士通株式会社岡谷貢様を始め，IPA 脅威と対策研究会構成員の皆様，亀山社中の皆様に深くお礼申し上げます．

本研究の一部は筆者が IIJ テクノロジーに所属している 2007 年から筑波大学との共同研究として行われているものです．共同研究をお許しいただいた菊池社長始め IIJ テクノロジーの皆様にお礼申し上げます．加えて本研究の一部は IPA の支援を受けております．ご支援いただいた IPA の皆様にお礼申し上げます．社会人学生として筑波大学への通学をご許可いただいたインターネットイニシアティブ齋藤衛様にお礼申し上げます．

業務を行いながらの研究を行う筆者をお気遣いいただき，また，叱咤激励いただいた業界関係者の皆様に感謝いたします．社会人学生という慣れない環境の中で実務面を支えていただいた岡本研究室の皆様，特に矢内様，誠にありがとうございました．

最後に，大学入学から本論文執筆まで筆者の研究活動を支えて続けてくれた妻に心から感謝します．