修士論文

SDNを利用した情報流出抑制システムに関する研究

A Study of the system to prevent information leakage using SDN

平成29年2月2日 提出

指導教員　関谷　勇司 准教授

東京大学大学院
工学系研究科　電気系工学専攻
37-146446　　佐藤　康次

要旨

標的型攻撃の被害が相次いでいる．特定のターゲットにしつこく攻撃をする標的型攻撃を，侵入を防ぐことを目的とする入口対策のみで守り続けるのは極めて困難である．ゆえに，侵入された後に被害を抑えることを目的とした出口対策の重要性が高まっている．情報流出を抑制する出口対策では，流出先の情報が必要である．流出先が判明していれば，通信を止めることで流出を抑制することができる．しかし，警告が出された脅威情報が全て危険だとは限らないため，脅威情報を元に一律に通信遮断することはできず，脅威情報を調査する必要がある．そこで，本研究では，脅威情報を自動解析しアクセス制御することで標的型攻撃による情報流出を抑制するシステムを構築することを目的とした．まず，脅威情報を解析するアルゴリズムを提案し，その流出抑制機能について検討した．検討の結果，提案アルゴリズムによって，大規模なデータベースを有する環境と限定すれば流出抑制機能があることを確認した．次に，提案システムを実現するにあたって SDN を用いることを提案し，その適切な実装方法について調査した．調査の結果，Packet In や Flow Stats などの OpenFlow で標準的な機能ではなく，トラフィックログを別途収集解析する手法が適していることが判明した．

Abstract

The damage of targeted threat is increasing.It is extremely difficult to prevent targeted threat by entrance countermeasures that aims prevent intrusion.Therefore, the importance of exit countermeasures aimed at suppressing damage after invasion is increasing.In exit countermeasures to suppress information leakage, information on destinations address is necessary.If the destination address is known, it is possible to suppress the leakage by disconnecting the network.However, since threat information for which warnings are issued is not always dangerous, it is not possible to uniformly disconnect communication based on threat information, and it is necessary to investigate the threat information.Therefore, in this research, we aimed to construct a system that suppresses information leakage by targeted threat by automatically analyzing threat information and controlling access.First, we proposed an algorithm to analyze threat information, and investigated its leakage suppression function.As a result of the examination, it was confirmed by the proposed algorithm that there is an leakage suppression function if it is restricted to an environment having a large scale database.And we proposed using SDN to realize the proposed system and investigated its appropriate implementation method.As a result of the examination, it turned out that a method to separately collect and analyze the traffic log is suitable, not a standard function in OpenFlow such as Packet In and Flow Stats.

目 次

図 目 次

表 目 次

第1章　序論

1.1　研究背景

標的型攻撃と呼ばれるサイバー攻撃による被害が相次いでいる (図 1.1)[1]．標的型攻撃とは，新しいタイプの攻撃 (Advanced Persistent Threat: APT)[2][3][4][5] と呼ばれる攻撃の一つであり，特定のターゲットに絞って執拗に行われるサイバー攻撃である．巧妙に偽装したメールからマルウェアに感染させる手法 (標的型メール) や悪意あるウェブサイトの閲覧によって感染させるなどの手法が利用される．例えば，2016 年 3 月に発生した JTB の個人情報流出事件では，オペレーターが標的型メールに添付されたファイルを開いたためにオペレーターの PC がマルウェアに感染し，JTB 組織内のオペレーター端末や Web サーバーにも感染が広がった結果，約 793 万人もの個人情報が流出の恐れがあると言われている [6].

このような標的型攻撃などのサイバー攻撃への対策は，組織ごとに対策をとる防御と脅威情報共有がある．組織ごとに対策をとる防御では，複数の防御層を組み合わせてセキュリティを高める多層防御が推奨されており，ネットワーク境界層や内部ネットワーク層，ホスト層などで様々な層で防御対策が行われている [7][8][9]．脅威情報共有は，脅威情報を共有し協力することで社会全体で対策することで，2015 年に経済産業省と IPA により勧告されている [10].

多層防御の各層における防御は，入口対策と出口対策に分けられる [11][12][13]．入口対策は，外部から侵入を試みる攻撃からシステムを守る対策手法で，Firewall や Intrusion Detection System(IDS)，ウィルス対策ソフトなどがあげられる．一方，出口対策は，入口対策をすり抜けてシステムに侵入されてしまった後に，情報の流出やシステムの破壊などの攻撃者の目的を達成させない防御であり，最重要部にインターネットが直接接続しないよう VLAN を設定することや利用者セグメントと管理セグメントを分離設計する [11] などがあげられる．標的型攻撃では特定のシステムに執拗に攻撃を繰り返すため，入り口対策で完全に防ぎきることは極めて困難である [14]．そのため近年では，出口対策の重要性が唱えられている [11][15]．出口対策に関する研究としては情報流出を防ぐことを目的とした

Data Leak Prevention(DLP) やトラフィック状況から攻撃を検知する研究などがあげられる [16][17][3]. DLP には，端末上でシステムコールを監視することで感染後にデータ流出を防ぐ対策 [16] やアプリケーションレベルで流出させないデータを設定し流出を防ぐ手法 [17] がある．そして，APT の攻撃を検知する手法として，ネットワークのトラフィックログから特徴を抽出し各ホストの感染状況をランク付けする研究 [3] などが研究されている．したがって，ネットワーク層の出口対策は，事前にネットワークを適切に分離設計することとトラフィックログから異常を検知するまでとなっている．

入口対策や出口対策を整える一方で，標的型攻撃などのサイバー攻撃への対応は，様々なステークホルダーを横断して解決に取り組むことが求められる [18][19][10]．組織間の連携のためには情報共有が必要であり，このための情報記述様式として Structured Threat Information eXpression(STIX)([20]) と呼ばれる脅威情報構造化記述形式が開発された．STIX を用いることにより，サイバー攻撃で観測された事象だけではなく，様々な脅威情報の交換を推進可能になる [21]. STIX などを利用して共有された脅威情報は，EC サイトのように多くの人からアクセスされることをが望ましいサイトなどでは，誤った情報で顧客との通信を止めるべきでないため，通信遮断をする前に情報解析が必要である．この際，誤検出を避けるため人によって判断を要求することが少なくなく，時間と労力を必要とする．しかし，情報セキュリティ人材不足とサイバー攻撃の高度化の中 [22]，人力で解析するのは限界がきている．そのため，脅威情報の解析をプログラムで自動化する必要がある．

情報流出を抑制する出口対策を実現するためには，流出先の情報を知り，流出先への通信を止める必要がある．流出先の特定ができていれば Firewall によって通信を止めることが可能だが，先述の通り脅威情報の中には危険だと断定されていない怪しい通信が存在するため，解析なしに脅威情報を利用して通信を止めることはできない．しかしながら，脅威情報を人力で解析するのは限界にきている．したがって，脅威情報を元にネットワークを監視し，脅威を判定するシステムが必要である．さらに，脅威判定ができていても通信を止めていなければ流出が発生するので，判定結果に応じてネットワークを制御するシステムも必要である．

1.2　研究目的

本研究では，脅威情報を自動解析しアクセス制御することで標的型攻撃による情報流出を抑制するシステムを構築することを目的とする．共有された脅威情報をプログラムで解析し，その結果を利用しネットワークのトラフィックの監視を行う．トラフィックの監視を

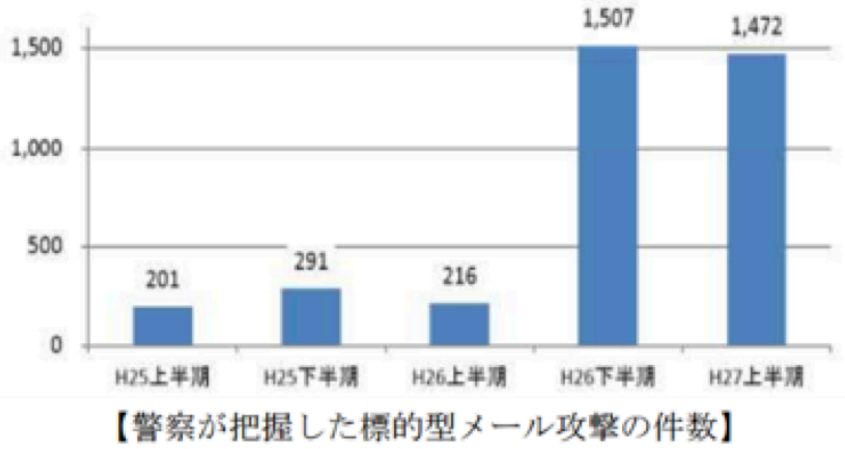

図 1.1　警察が把握した標的型メール攻撃の件数 [1]

しつつ脅威判定をし，危険と断定された通信を遮断する．このシステムを実現するために，内部ネットワーク層を Software Defined Networking (SDN) で構築することによって，ネットワークの監視と制御を行う．SDN でネットワークを監視と制御することにより，通信状況からその通信の脅威状況を解析したり，怪しい通信の通信経路を変更することでより厳しいセキュリティーチェックするなどが可能となる．このように，SDN によるネットワーク制御によって従来のネットワくシステムと比べて柔軟な対応ができる．さらに Software 制御であることから，STIX による情報共有を自動で解析し，ネットワークシステムに組み込むことが可能である．本研究のシステムにより，従来では人力が要求されていた脅威情報の解析とネットワーク制御の設定変更が自動化できる．

1.3　本研究の貢献

本研究で行なった検証実験により，提案システムを実現するための実装方法では，Data Plane と Control Plane という従来の OpenFlow に加え，Log Plane と Analyze Plane を加えるべきであることを示した．本研究の提案システムでは，トラフィックの状況を利用し脅威判定を行うため，トラフィックログが必要である．トラフィックログ管理を従来の OpenFlow で実装すると速度や情報量に不十分な点があった．そのため，既存の概念を拡張しトラフィックログ管理とネットワーク制御を分離した構造をとることで，従来の実装方法の不十分な点を解決した．

1.4　本論文の構成

本論文の構成は以下の通りである．第 2 章では，標的型攻撃の特徴や実際に発生した事例，標的型攻撃の対策手法や対策に関する研究について述べ，対策手法に置ける課題について議論する．第 3 章では，標的型攻撃の対策に必要な要件を明らかにしつつ，SDN を用いた内部ネットワーク層出口対策を提案し，さらに具体的に STIX によって共有された情報を元に，脅威度の判定と SDN によるネットワーク制御を利用した対脅威システムを提案する．第 4 章では，提案システムによる情報流出抑制の有効性についての評価実験と具体的な実装方法に関する調査実験の方法と結果について述べる．第 5 章では，実験結果についての考察と研究目的やシステム要件についての考察，提案システムの課題についての考察を述べる．最後に第 6 章で，まとめと今後の課題について述べる．

第2章　標的型攻撃

2.1　標的型攻撃について

新しいタイプの攻撃 (Advanced Persistent Thread: APT) は，ソフトウェアの脆弱性を悪用し複数の既存攻撃を組み合わせて特定企業や個人を狙った攻撃の総称 [3][4][11][5] であり，標的型攻撃はその一例である．APT では，複数の攻撃手法で複数の攻撃パターンで特定のシステムに攻撃をしかけるため，容易に攻撃を防ぐことができる防御手法はない．従って，多層防御によってシステムを守ることが基本となっている [7][8][9]．本章では，APT の中でも標的型攻撃に関して，攻撃手法や対策法について述べる．

2.1.1　標的型攻撃の攻撃手法

独立行政法人情報処理推進機構（Information-technology Promotion Agency: IPA) によると，標的型攻撃は 4 段階の攻撃に分類される [11]．初期潜入段階である第一段階でシステムに侵入し，第二段階で攻撃基盤を構築，第三段階でシステムの調査を行い，最後に第四段階で攻撃目的を遂行する（表 2.1）．この 4 つの段階のうち，第一段階と第四段階は，各攻撃によって攻撃手法が変わるが，第二段階と第三段階はどの攻撃でも共通的な攻撃手法が利用されている．

共通攻撃手法における機能を分類すると，ウィルスと攻撃者のサーバーとの通信を確立する http バックドア通信機能と，システム内の情報窃取の効率化のために多くの端末に感染させるシステム内拡散機能，システム内のウィルスに効果的な攻撃を行わせる機能をもたせるための一斉バージョンアップ機能，クローズ系システムの情報を収集するため USB 等にそのような機能のウィルスを入れ込む USB 利用型情報収集機能の 4 つに分類される．これらの機能は，内部から外部への通信は安全と考える従来のネットワークセキュリティ設計に起因している．共通攻撃を止めるための対策として，たとえ入口対策をすり抜けた場合でも攻撃者に情報を窃取させないことや重要システムを破壊させないことを目的とした出口対策 [11] の重要性が近年説かれている．

まとめると，標的型攻撃では，様々な手法でターゲットシステム内部に侵入した後，内部からの通信に対するセキュリティの弱さを利用してバックドアを作成する．そして，攻撃の効率化のために多くの端末に感染を広げつつシステム内部の調査を行い，攻撃目標の遂行を図る．

表 2.1　標的型攻撃の攻撃手法 [11]

段階	攻撃内容	特徴
第一段階 [初期潜入段階]	(1) 各種初期攻撃 ・標的型攻撃メール添付ウィルス ・ウェブ改ざんによるダウンロードサーバ誘導 ・外部メディア (USB など) 介在ウィルス	入り口の対策をすり抜け，システム深部に潜入 素早く次の段階へ移行 攻撃手法は使い捨て
第二段階 [攻撃基盤構築段階]	(1) バックドア (裏口) を使った攻撃基盤構築 ・ウィルスのダウンロードと動作指示 ・ウィルスの拡張機能追加	構築した攻撃基盤は発見されない 構築した攻撃基盤は再利用される
第三段階 [システム調査段階]	(1) 組織のシステムにおける情報の取得 (2) 情報の存在箇所特定	時間をかけて何度もしつこく行う
第四段階 [攻撃最終目的の遂行段階]	(1) 組織の重要情報 (知財・個人情報等) の窃取 (2) 組織情報 (アカウント等) を基に，目標を再設定	何度も攻撃を行うための情報窃取 組織への影響を与える情報窃取

2.2　標的型攻撃の事例

2.2.1　JTB の個人情報流出事件

2016 年 6 月 14 日，JTB は同社のサーバーが不正アクセスを受け，顧客情報が漏えいした可能性があると発表した [6][23]．公表された内容によると本事件では, 標的型メール攻撃と呼ばれる攻撃により JTB 社内の端末がマルウェアに感染し，攻撃者から遠隔操作を受けたことによって約 793 万人の個人情報が流出した恐れがある．

本事件の発端は，取引先を偽造して届いたメールに添付されていたマルウェアを JTB のオペレーターが誤って実行してしまったことによるマルウェア感染である．攻撃により感染したマルウェアは，標的型メールにより最初の端末に感染したあとに，別の PC 端末や Web サーバーにも感染を広げた．そして，最終的に何者かが約 793 万人分の個人情報を格納した CSV ファイルを作成し，隠蔽のために削除したとされる．

偽装されたメールでは，メールソフトに表示される From アドレスに実在する取引先企業ドメインが含まれる，社内でよく利用するファイル名を利用する，実在する取引先の会社名・部署と担当者の署名，exe 形式の実行ファイルを PDF ファイルに偽装した添付ファイルなど標的型メールだと気づかないように巧妙に偽装されていた．

2.3　標的型攻撃への対策

2.3.1　一般的な標的型攻撃への対策

標的型攻撃への対策では，入口対策と出口対策がある [11][12][13]．入口対策は，外からの攻撃を防ぐことを目的としている．例えば，インターネットとの境界で不正なアクセス元からの通信を遮断する Firewall や侵入を検知する Intrusion Detection System(IDS) があげられる．また，ホスト上でウィルス感染しているかスキャンするウィルス対策ソフトも入口対策である．反対に出口対策は，たとえ入口対策をすり抜けた場合でも，攻撃者に情報を摂取させないことや，重要システムを破壊させないことを目的とした対策である [11]．例えば，最重要部にインターネットが直接接続しないように VLAN を設定したり，利用者セグメントと管理者セグメントを分離設計するなどがあげられる．また，感染を検知して攻撃達成前に対処するという対策も，出口対策の一つと考えられる．しかし，現状では，JTB の事件のようにネットワークのログを外部セキュリティ機関が監視することで検知される状況である [10]．

2.3.2　脅威情報の共有

標的型攻撃などのサイバー攻撃への対応では，各システム内で対応を完結させずに適切に周囲に情報共有することも重要だ [18][21]．Structured Threat Information eXpression(STIX)[20] は，このサイバー攻撃に関する情報共有のために開発された．STIX は，サイバー攻撃活動 (Campaigns)，攻撃者 (Threat_Actors)，攻撃手口 (TTPs)，検知指標 (Indicators)，観測事象 (Observables)，インシデント (Incidents)，対処措置 (Courses_Of_Action)，攻撃対象 (Exploit_Targets) の 8 つの情報群から構成されており，xml などの記述形式で記述される (図 2.1[21], 図 2.2[21])．

このように，STIX は，サイバー攻撃対策において，検知に有効なサイバー攻撃を特徴付ける指標 (indicator) を含んだ情報である．したがって，STIX での情報共有は，サイバー攻撃によって観測された事象だけではなく，様々な脅威情報の交換を推進させる [21]．

2.4　関連研究

本節では，標的型攻撃などのサイバー攻撃による情報流出を抑える仕組みに関する研究について述べる．情報の流出自体を抑えることを目的とした Data Leak Prevention の研究，

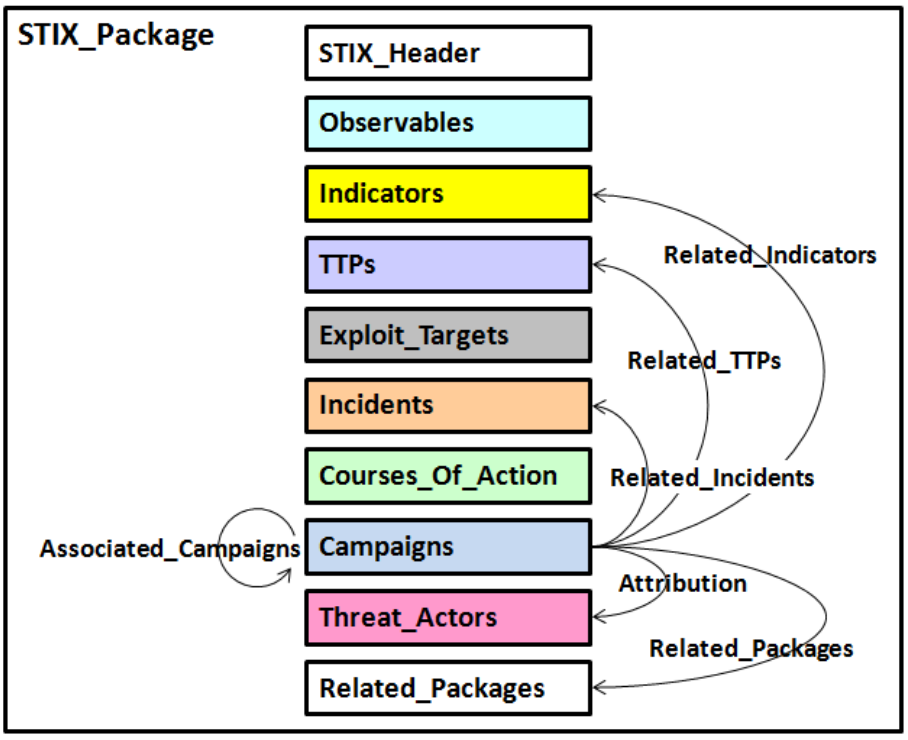

図 2.1　STIX 言語で記述する脅威情報の XML ファイルの構成 [21]

遠隔操作に利用される C&C サーバーとの通信を検知する研究，APT の攻撃を受けていることを検知する研究，SDN によるセキュリティに関する研究の 4 つに分類して説明する．

2.4.1　Data Leak Prevention

System Call Interception(SCI) を用いることで，情報流出に関係する Policy 違反行為を検出し対応することが可能である．Balinsky[16] らは，SCI を用いて DLP を実現するシステム (図 2.3[16])，を Microsoft Windows 上に実現した．このシステムは，動作のわからない Black Box なアプリケーションに対しても有効である．しかしながら，OS 毎にシステムを用意する必要があり，大規模な組織の端末全てに導入するのは困難である．

TaintEraser[17] は，アプリケーションによるネットワークまたはローカルファイルシステムへの意図しないデータ流出をブロックするツールである．キーストロークやファイルなどの監視する入力ファイルを指定し，ネットワークやファイルシステムなど指定したアウトプットチャネルへデータが移動するのを防ぐ．これを実現するために TaintEraser では，ネットワークとローカルファイルシステムへのアプリケーション出力を監視し，機密

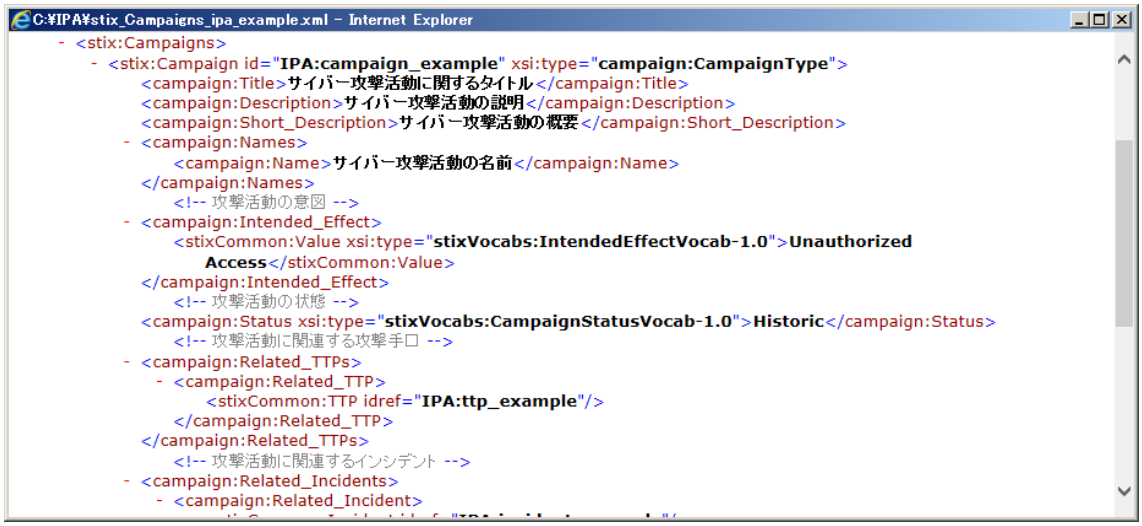

図 2.2　STIX の例 [21]

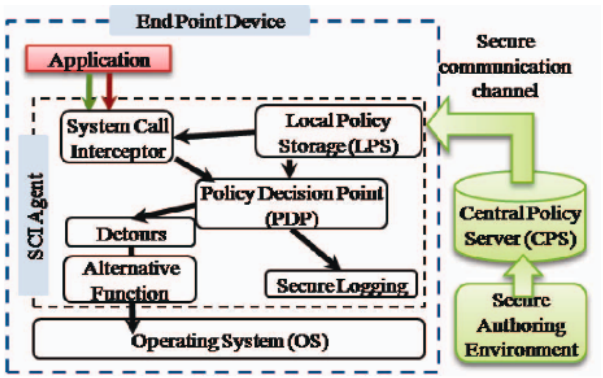

図 2.3　General architecture of the SCI framework[16]

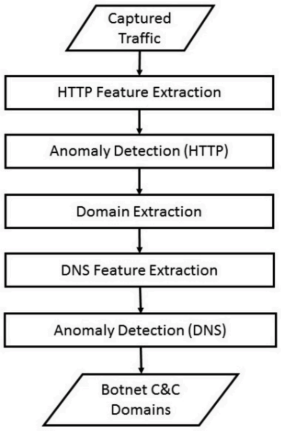

図 2.4　HTTP ベースの Botnet C&C トラフィック検知プロセス [25]

バイトをランダムなバイトに置換する．TaintEraser も Microsoft Windows を対象としている．

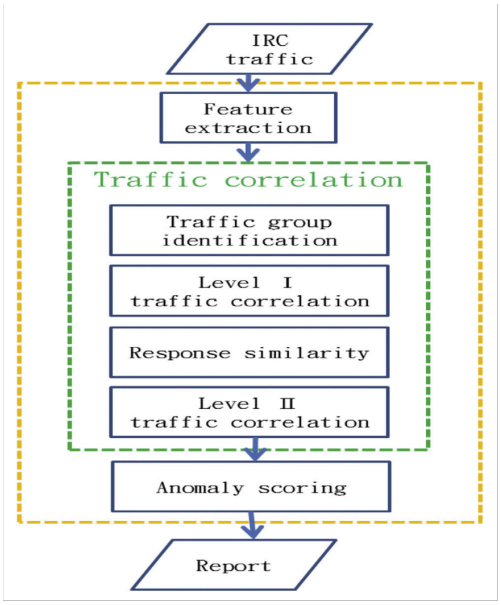

Table 1 – Flow attributes.

Feature name	Description
Sip	Source IP Address
Dip	Destination IP Address
Sport	Source Port
Dport	Destination Port
Time	Timestamp
Payload	Payload of IRC Traffic

図 2.5　検知に利用するパケットの特徴と検知システムのアーキテクチャー [24]

2.4.2　C&C サーバー通信検知

標的型攻撃の攻撃者は C&C サーバーを経由して感染端末の遠隔操作を行う．遠隔操作であるため，感染端末 (bot) と C&C サーバーはネットワーク通信を行なっており，その通信挙動から C&C サーバーとの通信を検知する研究がされている．

C&C サーバーから bot に対して指示を出すために，以前から Internet Relay Chat(IRC) がよく利用される．Chen ら [24] は，IRC で bot と C&C サーバーが通信する際の通信の類似性を利用することで，通信の異常度を算出することで bot となった端末と C&C サーバーを検知する手法を提案した．この手法で利用するパケットの特徴と，システムのアーキテクチャーを図 2.5[24] に示す．これらの特徴から bot と C&C サーバーとの通信との類似性を算出して異常度をだすことで，90%を超える true positive rate をだし，7%を下回る false positive rate を出した．

近年では，C&C サーバーとの通信には，C&C サーバーとの通信を通常の通信に紛れ込ませ検知しにくくするために，HTTP などのよく利用される通信プロトコルを利用される [15][25]．Muhammad ら [25] は，HTTP Request の特徴と DNS サーバーの Response の特徴を用いることで，HTTP を利用する C&C サーバーとの通信を検知する仕組みを構築した．HTTP Request の特徴は，ブラウザーが生成する HTTP Request とソフトウェアに

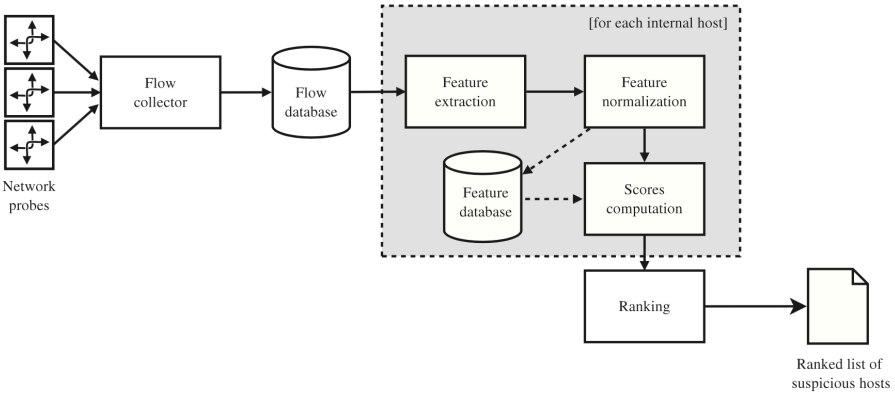

図 2.6　APT 検知の概要 [3]

よって生成される HTTP Request に分類し，DNS サーバーの Response の特徴では，正しい Web ドメインと C&C サーバーのドメインに分類し，これらの特徴を元にトラフィックに異常がないかを調査する (図 2.4[25])．Chebyshev ’s Inequality と One-class Support Vector Machine，Nearest Neighbor based Local Outlier Factor の 3 つの検知手法を用いて実験をし，Chebyshev ’s Inequality で約 94%もの検知率を出した．

2.4.3　APT 検知

パターンマッチングによる従来のセキュリティソリューションは，既知の攻撃検知には有効だが，新しい攻撃を利用したりよく利用されるプロトコルに紛れ込ませて通信する APT の検知は適していない [7]．また，従来のトラフィック解析技術は，Distributed Denial of Service(DDoS) や worms のような攻撃の検知は可能であるが，APT では熟練の攻撃者が一般的な動作に偽装するため，APT の検知は困難である [3]．Marchetti ら [3] は，巨大なトラフィックを収集し解析することで APT が関わるデータの流出の危険性をランクとして算出する手法を提案した．送信元 IP アドレスや宛先 IP アドレスなどのネットワークフロー情報を収集しつつ，ネットワーク内の各ホストでバイトサイズやフロー数，宛先数などの特徴を解析しランクを算出する (図 2.6[3])．約 10000 台のホストが接続された現実のネットワークで実証し，実現性と有効性を示したと報告されている．

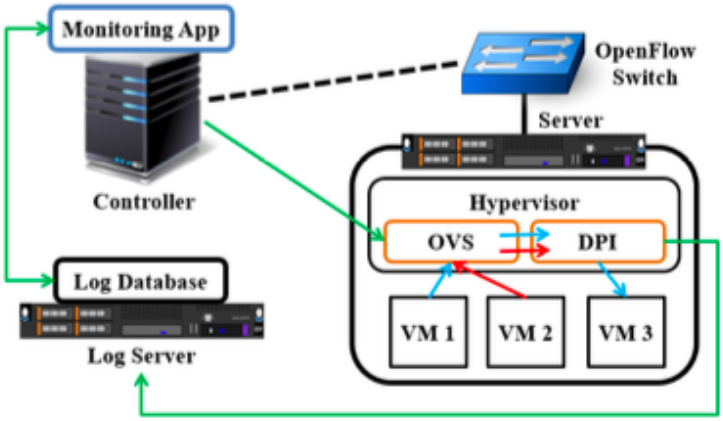

図 2.7　DPI を組み込んだ OpenFlow Switch の概要図 [28]

2.4.4　SDN によるセキュリティに関する研究

SDN のプログラマブルなネットワーク制御の特徴に着目して，SDN を使ったセキュリティの強化を目指す研究がある．Yoon ら [26] は Floodlight[27] を用いて SDN によって Firewall や IPS，IDS，ネットワークスキャンによる異常検知などのセキュリティ機能を実現した．OpenFlow の Packet In を利用し，Packet の解析を行うことでこれらのセキュリティ機能を実現している．Cho ら [28] は，OpenFlow で利用できるパケットヘッダーの情報だけでなく，Deep Packet Inspection(DPI) によってペイロードの情報を解析する仕組みを構築した．全てのパケットに対し DPI によるパターンマッチが行われ，マッチしたパケットの解析結果をログサーバに送られる (図 2.7[28])．ログサーバーの情報を元に Controller が転送判断を下す．

2.5　課題

2.5.1　出口対策における課題

標的型攻撃は特定のシステムに対して執拗に攻撃を繰り返すため，感染を防ぐことを目的とした入口対策対のみで防ぎきるのは困難である．そのため，感染後にデータ流出を防ぐ出口対策と組み合わせることが重要である [11][15].

出口対策では，感染したことを検知し感染端末のネットワーク遮断して攻撃を阻止するのを，攻撃終了までに行うことが求められる．そのためスピードが重要である．ただし，

端末によっては即時に利用不可にできない端末も存在するため，状況に合わせて柔軟に対応する必要がある．対策が反映されていない箇所を攻撃者に利用される危険性があるため，システムの一部ではなくシステム全体に対策が反映されていることも重要である．

2.4.1 項のようなホスト上での対策では，全ての端末にインストールしかつ環境に依存せずに実行させる必要があり完全な導入が困難である．また，感染した端末でこの仕組みが正常に動作する保証がない．したがって，このようなホスト上の対策も重要だが，システム全体に反映するのが困難なため，この対策のみでは不十分である．

2.3.1 項で述べたように，内部ネットワーク層の出口対策の一つとして，事前にネットワーク設計を適切に設定することが重要だ．しかし，日々巧妙になるサイバー攻撃に対し，通信状況に合わせた対応ができない柔軟性に欠ける事前設計による対策では十分に対策しきれない危険性がある．

2.4.2 項や 2.4.3 項，現状の外部セキュリティ機関による監視などを用いて感染などの脅威にさらされていることを検知するにはトラフィックのログが必要になる．従来のネットワークでは，トラフィックログを解析する機能がないため，これらの検知方式は全てネットワークの外部で実行される．故に，最短経路で検知しているとは言えない．さらに，検知結果をネットワークに反映するには人の手によってネットワークの設定を変更する必要があり，内部ネットワーク層内で対策が完結しない．そのため，対応が完了するまでに時間を要してしまう．

以上のことから，出口対策には，通信状況から脅威を判定した結果に応じて自動でネットワークを制御する機能がないという課題があげられる．

2.5.2　情報共有の課題

STIX などを利用して共有された脅威情報は，EC サイトのように多くの人からアクセスされることをが望ましいサイトなどでは，誤った情報で顧客との通信を止めると利益を損ねる危険性がある．また，システム内の多くの人が利用するサイトへの宛先が脅威情報に含まれた場合に，通信を遮断すると業務の支障をきたす可能性がある．このような環境では，誤検出や誤情報により業務を妨げたくないというニーズから，通信遮断をする前に情報解析が必要であり，セキュリティ運用の中で人の判断が関わることが多い．そのため，脅威の可能性はあるが断定が取れていない状態 (以下，怪しい) の情報では遮断などの対応はせず，その情報を解析し人が脅威だと断定した状態 (以下，危険) となってから対応が開始される．しかし，セキュリティ人材が不足しているなか [22]，日々大量に発生されるアラー

トに人力で対応するのは限界がきている．そのため，このような怪しい通信をプログラムが監視し脅威判定するように自動化する必要がある．

2.5.3　情報流出対策に対する課題

情報流出を抑制するには，流出先のアドレスを知り流出先への通信を遮断すれば良いので，流出先の特定ができていれば Firewall によって実現できる．しかし，2.5.2 の通り，脅威情報の中には危険だと断定されていない怪しい通信が存在するため，解析なしに脅威情報を利用して通信を止めることはできない．このように，脅威情報を元に通信遮断をするためには，その情報を解析する必要があるが，人力で解析するのは限界にきている．したがって，脅威情報を元にネットワークを監視し，脅威を判定するシステムを構築してサイバー防御を自動化する必要がある．このシステムや従来の攻撃検知手法などで脅威判定が実現された時に，人によってネットワークを設定し対応した場合，脅威情報解析と同じ課題に直面する．したがって，判定結果に応じて自動でネットワークを制御するシステムも必要である．

第3章　提案手法

3.1　標的型攻撃対策の要件

本研究では，脅威情報を自動解析しアクセス制御することで標的型攻撃による情報流出を抑制するシステムを構築することを目的としている．例えば，共有された怪しい通信先へある一定のアクセスパターンでアクセスした時に危険と判定し，通信遮断をしたり，従来研究を参考にトラフィックログから脅威を検知し遮断するなどが考えられる．また，一般的にセキュリティでは安全性と利便性はトレードオフの関係にあるので [29][30]，システムの実用性のために安全性と利便性のトレードオフの関係を考慮したシステムが望ましい．このようなシステムを実現するには，下記の 3 つの要件が必要である．

1. 怪しい通信先情報に対するアクセス制御の自動化
2. 情報が大量流出する前に流出を抑制
3. 安全性と利便性のトレードオフの関係を考慮

1 つ目の要件では，怪しい通信先情報を受け取り解析する機能の自動化とその解析結果に応じて通信遮断などの処理を行う機能が求められる．これにより，人力で解析していた怪しい通信先情報をコンピューターが自動で解析するようになり，さらにその対応まで人手を必要としなくなる．また，状況に応じた対応を自動でとることができるので，出口対策としての柔軟性が増し，遮断だけでなく段階的にセキュリティ機器と連携を強化するなど様々な対策が取れるようになる．2 つ目の要件は，情報抑制システムとしての機能である．守るべき情報が全て流出してしまう前に措置が取られることが求められる．3 つ目の要件では，可能な限り通常通りのネットワーク利用ができることが求められる．システムや状況次第で安全性と利便性の優先度は変わるので，このトレードオフが調整できることも要求される．通信遮断措置が取られた後に，その通信先に通信する必要が生じた場合や誤った判定だった場合に遮断措置が継続するのは利便性に欠ける．そこで，攻撃者ではない正規のアクセス者が，遮断されている通信と理解していることを確認した上で通信を復活させる機能を要件として設ける．

本研究では，上述のシステム要件を実現するにあたって，設計する際に柔軟性，素早さ，簡単さの特徴を有することを考慮した．日々進化するサイバー攻撃に対応するためには，様々な対策を自動で柔軟に取れることが必要である．そして，攻撃が達成する前に対処するために，対応開始から完了までが素早いことが求められる．ただし，様々な状況に合わせて柔軟に対応できる対策だったとしても，その設定が困難なために時間を要したり操作ミスの危険性が高い場合，実用性が低いと言える．

3.2　SDN を用いた内部ネットワーク層出口対策の提案

3.2.1　SDN を用いた内部ネットワーク層出口対策

3.1 節を踏まえて，本研究では，情報流出を抑制する出口対策として内部ネットワーク層の出口対策の強化と SDN で実現することを提案する．

出口対策では，ホスト上でネットワーク監視やデータの流出を防止する対策をとること可能である．しかし，2.5.1 項で述べたように，全ての端末への導入の困難性や感染端末の信用性の問題からホスト上での対策のみでは不十分である．遠隔操作や情報探索，感染の拡大，情報の転送など攻撃は全てネットワークを経由して行われるのでネットワーク上で出口対策をとるのは有効だと考えられる．しかし，Firewall など Internet との境界部分などで対応ではその内部の通信に対して対応することができない．また，端末から離れるため遅延や UDP の送信元 IP アドレスの偽造などの問題もある．そこで，本研究では，ホストや Internet との境界ではなく．内部ネットワーク層にネットワークの監視と制御のシステムを組み込むことを提案する．これにより，内部ネットワーク層の出口対策が強化され，より素早くより正確な対応が可能になる．

2.5.1 で述べたような通信状況から脅威を判定した結果に応じて自動でネットワークを制御する機能は，従来の内部ネットワーク層での出口対策 ([11]) では，ネットワークの事前設計のため実現できない．Firewall で通信状況を解析する機能を実現する場合，システム全体から流れるトラフィックが集中するため，負荷が高すぎる問題がある．そこで，SDN によってネットワークを構築することを提案する．SDN によってネットワークを制御することによって，脅威情報を自動で取り込み，怪しい通信の監視と制御が可能になる．また，SDN ではネットワークを構成する Switch 単位でトラフィックログ収集と解析が行えるため提案システムの分散が可能である．SDN ではネットワークをプログラムで制御するため，ネットワーク機器を人が設定し直す必要がなく，極めて簡単なオペレーションとなる．また，SDN のネットワーク制御を利用することで，怪しい通信先との通信だけ通信経路を変

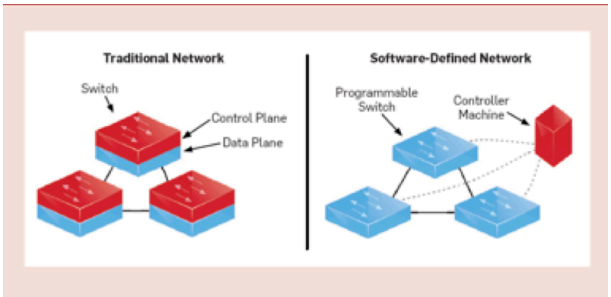

図 3.1　SDN の構造 [32]

えて，Firewall や IDS などのセキュリティ機器に通したり全パケットキャプチャするなどの従来のセキュリティソリューションとの連携が可能である．大学や大きな企業などの大きな組織では全ての通信を IDS に通したりパケットキャプチャするのは，通信量が多いため困難であり，さらに，IoT(Internet of Things) により，今後ネットワークを利用する端末が増加することが予想される [31]．したがって，怪しい通信先との通信のみに監視対象を制限するのは有効であると考えられる．

以上のことから，内部ネットワークの監視と制御を実現するために，SDN でネットワークを制御することを提案する．

3.2.2　SDN とは

SDN は，図 3.1[32] のように従来のネットワーク機器と違い，データプレーンとコントロールプレーンを分離し，ソフトウェアでネットワークを管理するという概念である [33]．従来の技術の場合，ネットワーク機器の内部にコントロールプレーンが存在し，それが転送先を決めデータプレーンがデータを転送している.SDN の場合，SDN Controller が転送先を決定するため，ネットワーク機器は指示通りにデータを転送する機能さえ有していれば良い．

SDN は，ソフトウェアでネットワークを管理するためプログラマブルという利点がある．このため，自動でネットワーク状態を変更するなどソフトウェアの性質を利用できる．また，そのようにプログラムでネットワーク構成を変更できるため物理的な設定変更が不要となり，柔軟で俊敏に対応できるという利点をもつ．図 3.1 のように Controller 一つで集中制御がするためネットワーク全体を一元的に管理することができる．これにより，トラフィック状況に応じた経路選択などに応用できる．VLAN では Layer 2(データリンク層)

のヘッダー ID を用いて管理するために仮想 LAN 数に制限があったが SDN ではそのような制限がなくスケールアウトが可能である．従来のネットワーク機器にはコントロールプレーンも搭載されていたため，ハイパフォーマンスを出すためには高価なものを利用する必要があるが，SDN の場合データプレーンのデータ転送機能があれば十分なためより安価なネットワーク機器が利用できるようになり，CAPEX が抑えることができる．また，柔軟性などの性質から運用が簡単なため OPEX も従来と比べて，低く抑えることができる．

まとめると，SDN には，プログラマブル，柔軟性や俊敏性をもつ，ネットワークを一元管理できる，制限がなくスケールアウト可能，コストなどの利点がある．

3.3　提案システム

本節では，3.2 節を踏まえて，脅威情報で得られた怪しい通信先の情報を元にネットワークの監視と制御を行うことによって，情報の流出を抑制するシステムの概要を説明する．監視対象の怪しい通信先の情報は STIX によって入力される．入力された STIX を解析し，その情報を元に IP アドレスなどの監視項目を割り出す．そして，監視項目に該当するアウトバウンド方向の怪しい通信に対し，脅威指標や情報流出の危険性を元に通信遮断の対応をとる．この様子を図 3.2 に示す．

怪しい通信先への通信は，怪しい通信先とのトラフィックログを利用する脅威判定システムによって脅威指標を判定して脅威指標の高い通信は遮断する．ただし，怪しい通信先にデータを送る必要が出た際に通信するために，アクセス者による確認によって遮断された通信先へ通信を復活可能なシステムとする．そこで，SDN コントローラーと連携した確認システムを利用することによって，通信サイズによって遮断された通信を通す仕組みを用意する (図 3.3)．脅威判定により危険な通信と判断された通信は，SDN の制御によってシステム内に置かれている確認サーバーに通信を転送される．確認サーバーと通信することになったホストは，システムで定められた認証を通すことによって，SDN コントローラーによる確認サーバーへの転送を解除できる．

図 3.2 における脅威判定システムの一例を図 3.4 にあげる．この脅威判定システムは，**?? 節で後述する Flow Stats を利用した実装でも実現可能なアルゴリズムとして通信回数を用いる．Indicator で示された怪しい通信先に対して，アウトバウンドに通信する際にその通信先との通信状況での判定をする．Indicators に記された IP アドレスとの通信回数を a_1, a_2,...,a_n ($n > 0$ の整数)，ドメインを u_1, u_2,...,u_m ($m > 0$ の整数) とする．このとき，怪しい通信先との総通信回数 ($\sum_{i=1}^{n} a_i + \sum_{i=1}^{m} u_i$) が多いほど脅威指標が高くなると考えられ

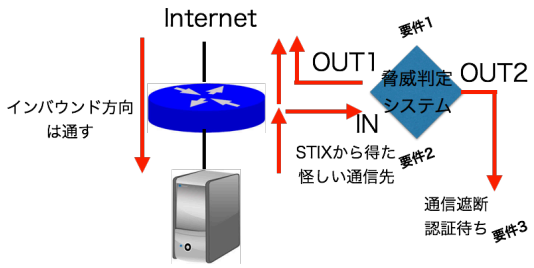

図 3.2　システム概要

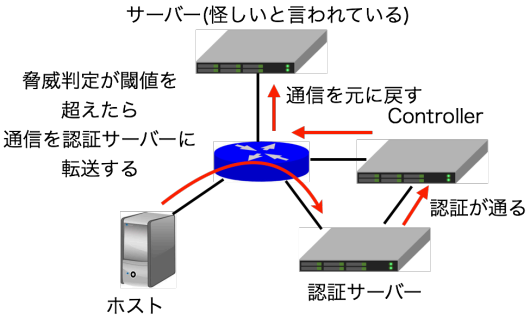

図 3.3　SDN による確認システム

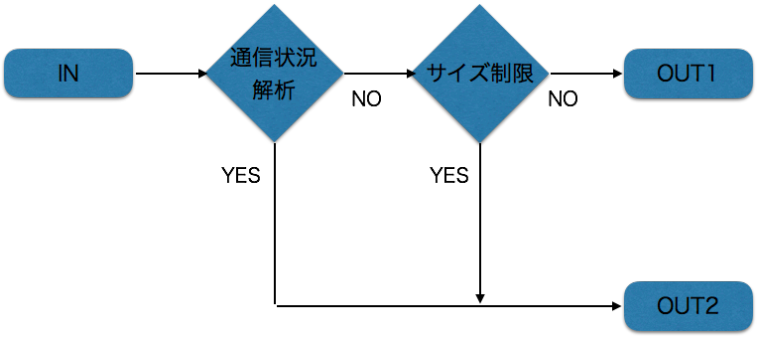

図 3.4　怪しい通信先への脅威判定システムの例

る．また，n 個の IP アドレスのうち N_a 個と通信をしていて，m 個のドメインのうち N_u 個と通信をしている場合の，Indicators との一致率を $C(=\frac{N_a+N_u}{n+m})$ とする．C の値が大きいほど，STIX で共有された怪しい通信先との通信をしていることになるので，脅威指標が高くなると考えられる．以上のことから，脅威指標 (T) を下記の式から判定できると考えられる．

$$T = C \times (\sum_{i=1}^{n} a_i + \sum_{i=1}^{m} u_i) \tag{3.1}$$

脅威指標判定で通信許可が降りた通信を監視しているだけでは，脅威判定をすり抜けた攻撃の場合に情報の流出を許してしまう危険性がある．そのため，脅威判定で許可が降りた通信だとしても，情報流出する可能性の高い通信は遮断するべきだと考えられる．情報が流出するときは，アウトバウンドにサイズの大きい通信が発生する．反対に，ブラウザで Web ページを見るときなどの通信はアウトバウンドの方向の通信は GET リクエストを

飛ばすだけなので通信サイズは大きくない．故に，ファイルのアップロードなどを除くと通常の利用では通信のサイズは限定されると考えられる．そこで，通信のサイズの大きさによって情報流出する危険性の判定を行う．

個人情報の流出と仮定すると，名前や電話番号，住所などの情報となるため，一人当たりのデータサイズを S_p とおく．通信制限のサイズを S_l とすると，一人の情報を流出させるのにかかる通信回数は $N_l = \frac{S_p}{S_l}$ となる．ここで，流出する際に最も対応が厳しい状況は Indicators の指標のうち一つしか該当していない時である．つまり，脅威指標の閾値 (T_{th}) は下記の式を満たす必要がある．ただし，a は Indicator の指標のうち唯一該当している通信先との通信回数である．

$$T_{th} \leq \frac{a}{n+m} \tag{3.2}$$

また，流出人数を N_p 人までに抑えることを考えると，a 下記の条件を満たす必要がある．

$$a \leq N_l \times N_p \tag{3.3}$$

よって，脅威指標の閾値 (T_{th}) の条件は以下のようになる．

$$T_{th} \leq \frac{N_l \times N_p}{n+m} \tag{3.4}$$

通信許可が下りる通信では，$T \leq T_{th}$ を満たすので，3.1 と 3.4 より

$$C \times (\sum_{i=1}^{n} a_i + \sum_{i=1}^{m} u_i) \leq \frac{N_l \times N_p}{n+m} \tag{3.5}$$

となり，3.5 の両辺に $n+m$ をかけると

$$(N_a + N_u) \times (\sum_{i=1}^{n} a_i + \sum_{i=1}^{m} u_i) \leq N_l \times N_p \tag{3.6}$$

となる．3.6 の左辺は怪しい通信先との通信回数を表しているため，怪しい通信が増えるとこの式の条件が満たせなくなり，通信が遮断される．

3.4　SDN の実装方法

3.4.1　実装要件

本研究では，SDN を実現するにあたって最も一般的な OpenFlow プロトコル [34] を採用し，Switch は Open vSwitch[35] を利用した．本節では，この環境における提案システム

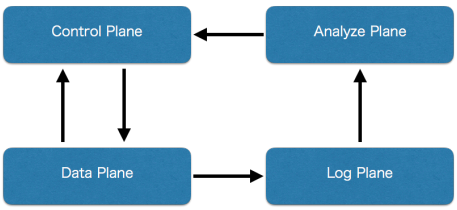

図 3.5　セキュリティに向けてトラフィックログ収集もとらえた SDN

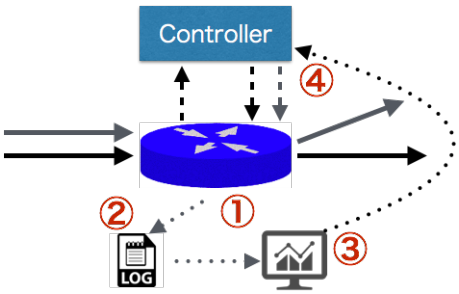

図 3.6　OpenFlow を拡張し，トラフィックログ管理とネットワーク制御を分離した実装方法

の実装要件について説明する．提案システムでは，怪しい通信先との通信状況をもとに脅威判定を行い，その結果に応じて対応を変える．したがって，怪しい通信先とのトラフィックログが必要である．ゆえに，本研究での SDN の実装要件は，トラフィックログの管理が可能で，そのログ情報をもとにネットワークを監視し制御できることである．

3.4.2　実装方法

OpenFlow でトラフィックログを利用する場合，Packet In を利用するか Flow Stats を利用するかのどちらかになる．しかし，これらの方式では，Controller に負荷が集中してしまったり，トラフィックログ情報が統計情報として欠如してしまう問題があり，実装要件やシステム構築に不適切である．これは，OpenFlow がトラフィックログの収集を考慮していなく，実装するにはネットワーク制御の実装にトラフィックログ管理機能を同居させなければならないことが原因である．そこで，OpenFlow を拡張してトラフィックログ管理とネットワーク制御を分離することを提案する (図 3.5)．本研究では，OpenFlow や Open vSwitch を直接拡張する前段階として，トラフィックログの収集に tcpdump と MongoDB[36] を用いる．まず，怪しい通信先との通信に対し，ミラーリングによってパケットを複製し本来の転送先ポートとは別に特定のポートに出力し tcpdump する ①．次にその pcap ファイルを定期的に JSON に変換して MongoDB に保存する ②．そして，Analyze Plane として用意したプログラムが定期的に MongoDB を参照し ③，監視対象となる怪しい通信先との通信ログを元に脅威指標を算出する．脅威判定の結果危険な通信と判定された場合，Controller に対して REST API 経由で通信をし，通信を受けた Controller は該当する通信を遮断もしくは確認サーバーへ転送するように Switch に指示を出す ④．この様子を図 3.6 に示す．

第4章　評価実験

4.1　予備検討

4.1.1　SDN による出口対策の有効性検討

本項では，SDN で内部ネットワーク層の出口対策を実現する有効性の検討を目的として，提案手法のシステムと従来のネットワークシステムを比較する．3.1 節で述べた素早さ，柔軟性，簡単さの 3 要素に関して，提案手法と従来のネットワークシステムでの対応方法における特徴について論じる．従来手法としては，感染端末を LAN を抜いたり無線を無効にするなどの物理的にネットワークから遮断する方法 (A) と Firewall のみで対応する手法 (B)，Firewall とその他のネットワーク機器で通信制御する方法 (C) の 3 パターンと比較する．

速さ

感染後の対応である出口対策では対応開始から完了までを素早く行う必要がある．ここで対応開始のタイミングは不審な通信などで感染が判明した時とする．対応完了は，攻撃者との通信と流出を防ぐために外部ネットワークとの通信制限に加え，感染の拡大や情報探索を防ぐために内部ネットワークとの通信制限をした時とする．

(A) の対応時は，感染端末を直接操作するため，感染端末の近くにいる人に指示を出すか感染端末のある場所に赴く必要がある．さらに，人が操作することを考慮すると遅い対応だと言える．(B) の対応時は，Firewall で外部との通信を制限するのはネットワーク管理者によって即時に対応が可能である．しかしながら内部の通信に関しては Firewall より内部の制限はできず，即材に対応することはできない．(C) の場合でも Firewall より内部での通信制限をかける場合は直接 Switch などを操作する必要がある．従って，外部通信の制限は即時対応できるが内部通信の制限に関しては環境によっては時間がかかると言える．従来手法では，SDN コントローラーから感染端末近傍の Switch に対して通信制限の命令を出すことで外部とも内部とも通信制限が可能であるため，即時対応できると言える．

簡単さ

感染を検知した後の対応は簡単であることも重要である．対応が困難であると，時間がかかったりミスをしてしまう可能性があるためである．さらに，困難性から対応できる人が限られてしまうと人的コストや即時対応ができないなどの問題も考えられる．

(A) では，IP アドレスの管理がされていれば，端末を特定してそのネットワークを遮断するだけのため簡単である．しかし，DHCP を利用しており IP アドレスが管理されていない場合，端末の特定が必要になり調査にネットワークの知識を要し，若干の手間がかかると考えられる．(B) と (C) でも，(A) と同様に IP アドレスが管理されているかで簡単さが変わるが，(B) と (C) の場合，ネットワーク機器で対応する技術と知識が必要になる．さらにネットワーク機器は製品や環境によって操作方法が違うため状況によって対応が変わることになる．従って，(B) と (C) での対応は簡単とは言えない．SDN の場合，IP アドレスと端末のマッピングを行っていれば，SDN が自動で端末を特定でき，対応も SDN が仕様に基づいて処理をする．要するに，SDN コントローラーに指示を出すだけで良い．

柔軟性

Web サーバーなど感染端末によっては即時にネットワーク遮断やシャットダウンなどで利用不可にできない端末も存在するため，一部通信を制限するなど状況に合わせて対応する必要がある．従って，対応手法は単一の状況にだけ対応できるのではなく，様々なパターンに柔軟に対応できることが重要である．

(A) の対応では，感染端末を物理的にネットワークから遮断するため，通信遮断以外の対応ができなく，柔軟な対応ができるとは言えない．(B) では，Firewall のみでは外部通信との制限程度しか対応できなく柔軟性にかける，これに対し (C) では，ネットワークを構成するネットワーク機器の機能を利用することで，内部通信の制限など柔軟に対応できる．ただし，ネットワーク機器に機能が実装されていて，各機器に設定することが必要とされる．しかしながら提案手法では，SDN によってコントロールプレーンをソフトエウェアで制御しているため，利用しているプロトコルや機器の機能の範囲内であれば，より柔軟に対応できる．

ネットワークでの攻撃への対応は，下記の 6 つがあげられる．これらの機能と対応手法との関係を表 4.2 に示す．

(i) 感染端末のネットワーク遮断

(ii) 特定の IP アドレスとの通信を遮断

(iii) 特定のポート通信のみ許可

(iv) 内部通信の遮断

(v) 感染端末が通信した内部通信の特定

(vi) 特定の通信フローのみ許可

表 4.1　出口対策で要求される機能と対応手法の関係

	(A)	(B)	(C)	提案手法
(i)	○	×	○	○
(ii)	×	○	○	○
(iii)	×	○	○	○
(iv)	×	×	○	○
(v)	×	×	○	○
(vi)	×	×	○	○

さらに，SDN を用いたネットワークでは，ソフトウェアで制御する特性から従来のネットワークでは実現できない機能を有することができる．例えば，

(i) 怪しい通信のみ通信経路を変更して監視の強化

(ii) 他のネットワーク機器の情報と組み合わせた解析

(iii) 通信に対する認証システム

が挙げられる．

総評

4.1.1，4.1.1，4.1.1 をまとめたものを表 4.2 に示す．○は対応として優れていることを示し，△は基本的には優れた対応だが状況や環境によっては評価が下がることを示し，×は良くない対応であることを示している．

(A) は状況によっては簡単な対応であるが，対応完了までの速さや柔軟性に欠けており良い対応方法とは言えない．(B) は素早く対応することは可能だが，対応可能な機能が限

定されてしまい (B) のみで対応するのは不十分だと考えられる．(C) は時間と手間をかければ様々な対応ができる．しかし，出口対策では流出する前に対応する必要があることを考慮すると，時間がかかりなおかつ対応する人物に技術と知識を要求する対応方法では最適とは言えない．ただし，2 次対策として (C) の手法を利用して，より強固な対応をするのは効果があると考えられる．提案手法は，対応開始から完了までの速さと対応の簡単さ，状況に合わせて様々な対応を取れる柔軟性を有しており，優れた対応手法だと言える．

表 4.2　ネットワーク上での出口対策に関する特性比較

	速さ	簡単さ	柔軟性
(A)	×	△	×
(B)	○	△	×
(C)	×	×	○
提案手法	○	○	○

4.1.2　STIX 調査

本項では，実際に利用されている STIX を解析し，その特徴を調査した．解析には，JPCERT から共有していただいた 30 ファイルの STIX を利用した．

STIX の 8 つの Field の利用頻度を調査した結果を表 4.3 に示す．Indicators は 30 ファイル全てに記載されており，Incidents は 26 ファイル，TTPs は 4 ファイルでその他の Field は記載されていなかった．このことから，現状の STIX では 8 つの Field が用意されているが，それらを使いこなせていないということがわかる．本調査においては，Indicators は全てのファイルに記載されていたが，これは Indicator という検知指標の記述が STIX の特徴の一つ ([21]) だからと考えられる．したがって，現状の STIX においては，Indicator 以外の情報はなくても影響のない補助的な情報として扱うべきである．また，Indicators 以外の情報は，脅威に対するタイトルや説明文と関連する攻撃の ID であった．ゆえに，現状の STIX の利用方法のままでは，コンピューターの解析によって脅威情報を的確に判断するのは困難だと考えられる．

次に，Indicator の中身について調査した．本研究では，Indicator の中でも IP アドレスやドメインの情報を利用するので，それらの記載状況について調べた．その結果，30 ファイルには 1 ファイルあたり平均 13.9 個の IP アドレスかドメインの情報が記載されていた．

表 4.3　STIX の Field の利用頻度

Field 名	利用回数
Observables	0
Indicators	30
TTPs	4
Exploit Targets	0
Incidents	26
Courses of Action	0
Campaingns	0
Threat Actors	0
Related Packages	0

4.2　脅威判定アルゴリズムの検討

本節では，3.3 で提案した脅威判定アルゴリズムについて情報流出抑制の効果があるか検討する実験について説明する．

4.2.1　情報流出の恐れのある通信に対するサイズ制限

実験の目的と内容

制限サイズの参考とするために通常利用の通信サイズの収集を行った．情報が流出する方向への通信として，内部の端末から外部へ向かう通信の様子を調査した．提案システムでは，通常の利用を阻害しないことを目的としているため，最も一般的に利用されているツールとしてインターネットブラウザを想定し，HTTP をターゲットとした．ただし，HTTP の POST 通信によるファイル転送はバラツキが大きく通常の利用の指標に不適切なのでデータから除き，HTTP GET Request 通信フローのサイズを計測した．実験環境を図 4.1 に示す．ラズペリーパイを NAT ルーターとして設定し，無線アクセスポインタと接続する．この無線アクセスポインタにノートパソコンを接続することで，ノートパソコンとインターネットとの通信をラズペリーパイを経由させ，tcpdump を利用して 80 番ポートへの通信ログを取得した．ノートパソコンは同時に最大で 4 大接続し，60 時間収集した．ただし，60 時間常時データを収集したため，夜間帯などノートパソコンと接続されていない時間帯も存在する．

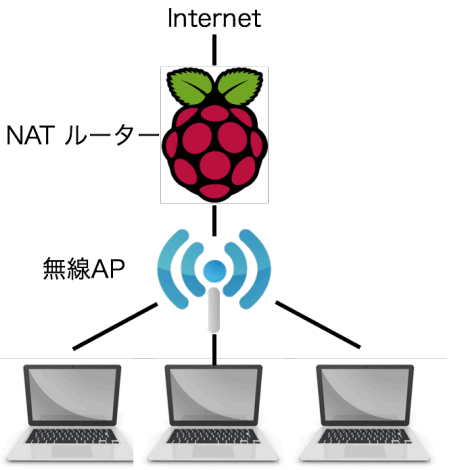

図 4.1　HTTP GET Request サイズ集計の実験環境

実験結果と考察

外部への HTTP GET 通信のサイズ分布の累計グラフを図 4.2 に示す．外向きの HTTP GET 通信の通信回数は 7191 回であり，そのうちの 90%はおおよそ 2100 バイト以下であり，80%は 1200 バイト以下，70%は 800 バイト以下であった．

2000 バイト近くの通信の内容を調査したところ，Cookie によるログイン情報を保持した通信が多かった．ログインした状態でページを回遊し，さらに定期的に閲覧することから 2000 バイト付近の通信回数が増加していると考えられる．

したがって，普段のインターネット利用で生じる外向きの通信のサイズは，たいていの場合は 1 パケットで済む小さな通信だと考えられる．

4.2.2　怪しい通信先との通信許可数と流出人数の関係

実験の目的と内容

実験 4.2.1 の結果と式 3.6 用いて，提案システムの通信サイズ制限と脅威度判定の有効性を調査する．

まず，本実験では情報流出を個人情報の流出に絞る．さらに，日本年金機構や JTB の個人情報流出事件から個人情報を氏名，性別，生年月日，住所，電話番号と仮定する．日本人の苗字の平均文字数は 2.18 文字である [37]．名前の平均文字数は，明治安田生命が公開している名前ランキングの生まれ年別ベスト 10[38] を元に計算したところ 1.85 文字であっ

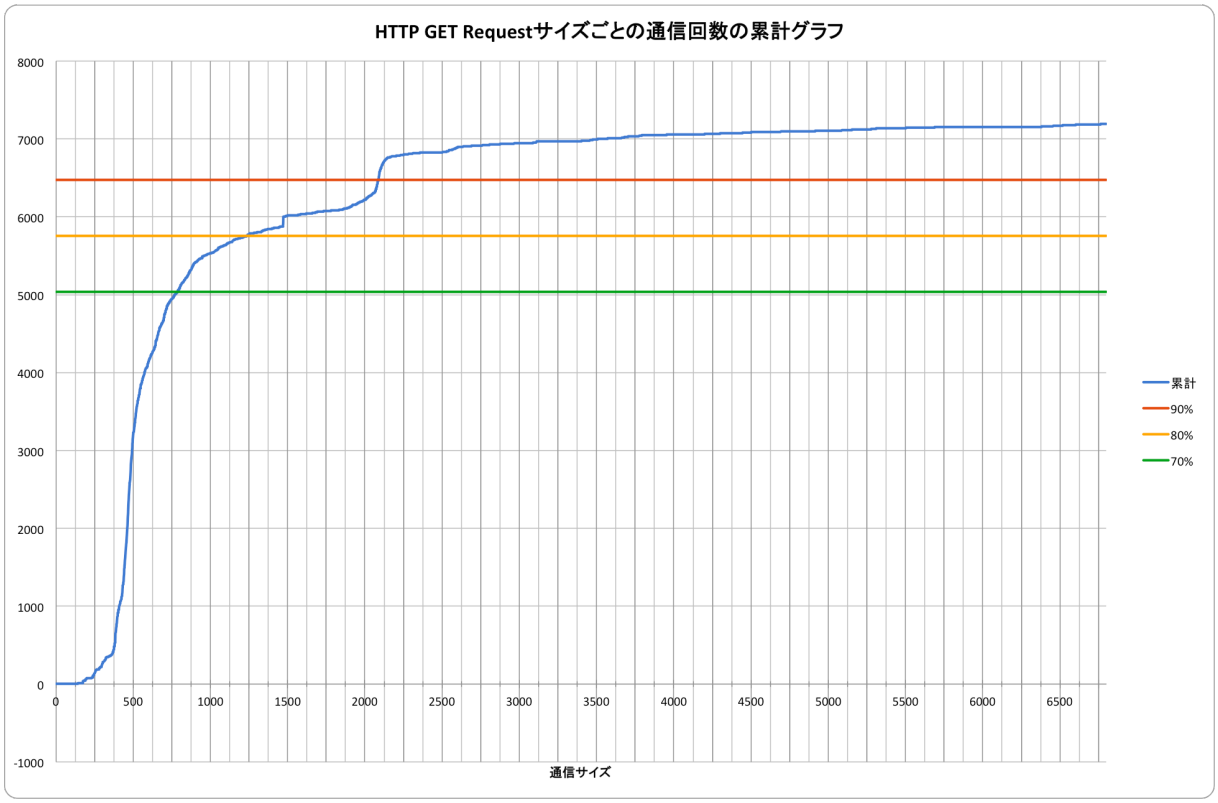

図 4.2　HTTP GET Request サイズ毎の通信回数の累計

た．住所では，日本郵便配布の住所データ [39] を元に計算したところ，都道府県名と市区町村名，町域名までの平均文字数は 11.1 文字であった．UTF8 で日本語 1 文字 3 バイトとした場合の個人情報の平均サイズの様子を表 4.4 に示す．これらから，本実験では一人の個人情報は平均 111.4 バイトであると仮定する．

本実験では，この数値と実験 4.2.1 を結果を式 3.6 に代入することで，怪しい通信先への通信遮断を決定する通信回数と流出する人数の関係を調査する．

実験結果と考察

実験 4.2.1 を参考に，通信サイズ分布の 90%(2100)，80%(1200 バイト)，70%(800 バイト) の通信制限を実施した場合の流出人数の様子を図 4.3 に示す．なお，図 4.3 には参考のためにイーサネットにおける 1 パケットの最大サイズの 1500 バイトでのグラフも図示している．横軸は式 3.6 の左辺の値に該当し，怪しい通信先への通信が許可される回数に相当す

表 4.4　ターゲットとする個人情報とその平均サイズ

項目	平均サイズ (バイト)
氏名	12.1
性別	3
生年月日	33
住所	33.3
電話番号	30

る．また，縦軸は流出人数 (N_p) である．したがって，図 4.3 は怪しい通信先への通信遮断を開始する回数と，遮断するまでに流出する個人情報の人数の関係を示したグラフである．

グラフより，通信サイズ分布の 90%の通信制限をかけた場合，約 1 万 9 千人の流出で抑えられる．同様に，80%では約 1 万 1 千人，70%では約 7 千人である．日本年金機構の個人情報流出事件では 125 万人，JTB の個人情報流出事件では 793 万人の個人情報が流出した危険性があると言われている．これらの事件のような大規模なデータをもつ組織では，提案システムは情報抑制効果がある可能性がある．また，許容する通信回数を変化することによって安全性と利便性のトレードオフを調整することができる．

イーサネットにおける 1 パケットの通信サイズは最大で 1500 バイトである．実験 4.2.1 の結果では，1500 バイト以上の通信が 15%程度存在する．このような大きい通信は Cookie などでログインした通信が多数を占めていた．ログイン情報を怪しい通信先に送信するのは，その情報を元に被害が拡大する危険性がある．ここで，怪しい通信先に対して提案システムによって通信遮断の措置が取られた場合でも，アクセス者確認によって通信を復活させることが可能である．よって，1500 バイトより大きい通信，つまり 1 パケットに収まらない通信は遮断で良いと考えられる．これにより，トラフィックフローではなくパケットごとに通信サイズ制限をかければ良いと言える．パケットごとに通信サイズの制限をする場合，1500 バイト以下になるため図 4.3 では 80%と 70%のグラフが該当する．

4.3　OpenFlow での実装についての検討

本節では 3.4.1 節で述べた実装要件に適した実装方法についてプロトタイプを作成し比較検討した．本実験では，OpenFlow を利用した方式として Packet In を利用した方式 A と Flow Stats を利用した方式 B の 2 つの方式についてスループットとそれぞれの特徴の定性評価をもとに評価した．

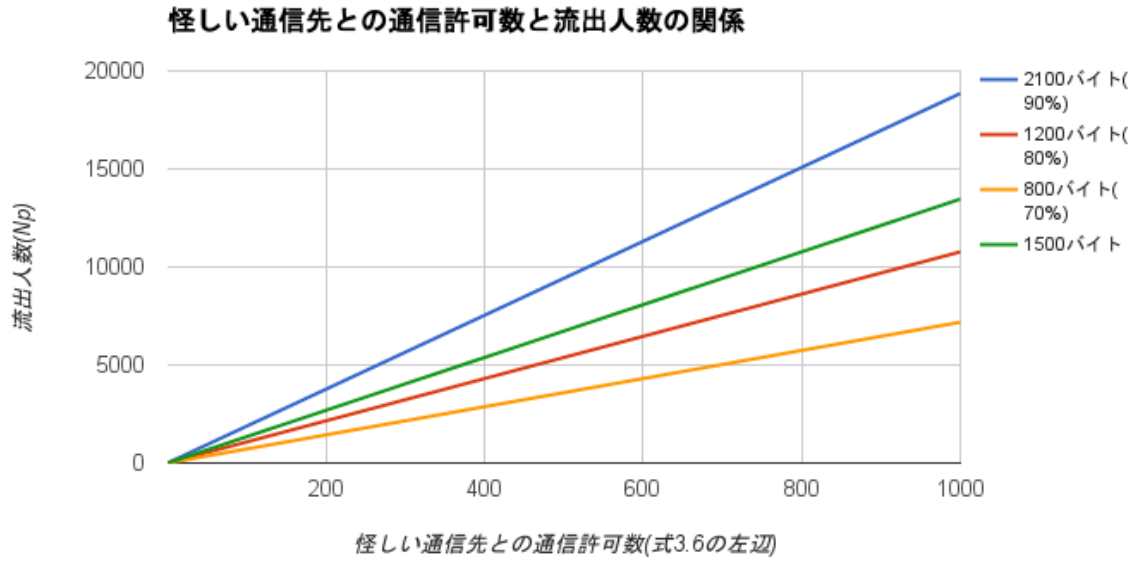

図 4.3　怪しい通信先との通信許可数と流出人数の関係

4.3.1　プロトタイプ

Open Flow による実装

方式 A, B の実装方法の解説の前に Open Flow による実装について解説する．OpenFlow では，Switch がに Flow Table と呼ばれる Flow Entry(パケットに対する条件や処理，統計情報) を管理するデータベースが存在する．Switch にパケットが入ると，この Flow Table の条件にマッチするかを判定し，条件にマッチする Flow Entry があったらその処理に従う．マッチする Flow Entry がない場合，Switch は Controller にパケットの転送先を確認するために Packet In という通信を流す，Packet In を受け取った Controller はプログラムに従ってパケットの転送先を決め Packet Out で Switch に転送先を支持する．この様子を図 4.4 に示す.

Flow Entry には Flow Stats というパケットが通った回数やパケットのサイズなどの統計情報のデータが保持されている．OpenFlow では，Controller が Switch に対して Flow Stats Request を飛ばすと Switch が統計情報を Flow Stats Response として返す仕様となっている (図 4.5.

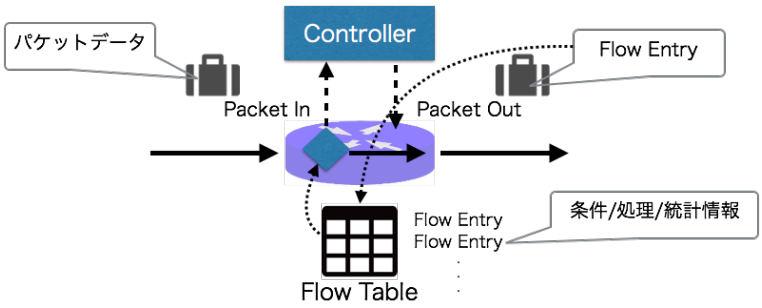

図 4.4　OpenFlow による転送処理の様子

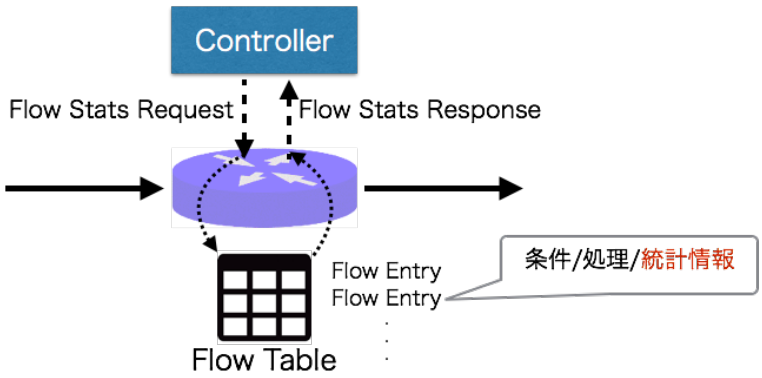

図 4.5　OpenFlow による Flow Stats の仕組み

方式 A: Packet In を駆使する

Packet In を駆使する方式 A では，まず怪しい通信先との通信を全て Packet In させる．Packet In で Controller に渡された怪しい通信先とのパケットをトラフィックログとして管理する．そして，そのトラフィックログを元に脅威判定を実施し，判定結果を元に転送先を決め Switch に指示を出す．この方式を用いることにより，トラフィック管理がされ該当するパケットにリアルタイムに解析結果を反映できる．

方式 B: Flow Stats を利用する

Flow Stats を利用する方式 B では，まず怪しい通信先との通信の Flow Entry を登録する．そして，定期的に Controller が Switch に対し Flow Stats Request を飛ばし，怪しい通信先との通信回数の情報を取得する．取得した通信回数の情報を元に脅威判定を実施し，判定結果を元に転送先を決め Switch に指示を出す．この方式では，トラフィックログが統

計情報に丸められてしまい情報量が低下してしまうが，方式 A に比べ Controller への負担を少なくできる．

4.3.2　実験環境

本実験は，サーバー上に mininet で仮想ネットワークを構築し，nuttcp[40] を用いてネットワークのスループットを調査した．OpenFlow プロトコルを採用し，Ryu フレームワーク [41] で SDN コントローラーを実装した．実験環境の詳細を表 4.5 に示す．

表 4.5　実験環境

項目	version
OS	Ubuntu 16.04 LTS
Ryu	4.9
OpenFlow	1.4
Open vSwitch	2.5.0
mininet	2.3.0d1
nuttcp	6.1.2

4.3.3　実験 1: 最もシンプルな条件でのスループット

実験の目的と内容

2 つのプロトタイプの性能比較の第一段階として，最もシンプルな条件でのスループットを調査した．Host1 台，Switch1 台，Controller1 台という最もシンプルなトポロジーのネットワークを構成し，怪しい宛先を 1 つ監視対象とした (図 4.6)．通信先のサーバーは mininet を起動している端末 (②) とし，mininet 上の Host(①) から ② へのスループットを nuttcp で測定した．

実験結果と考察

図 4.7 に各プロトタイプにおいて怪しい通信先に通信した場合と普通の通信先に通信した場合のスループットを 10 回計測した平均を示す．なお，グラフのエラーバーは標準偏差を利用した．グラフより，方式 A の Packet In 方式で怪しい通信先に通信する場合に，極端にスループットが下がっている．方式 B は，このシンプルな条件では通信速度に影響は

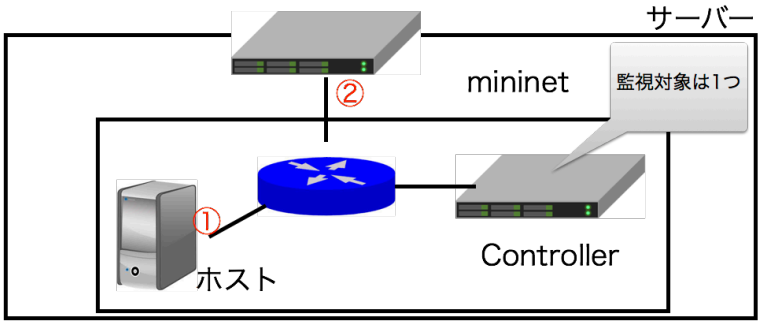

図 4.6　実験 1 の概要図

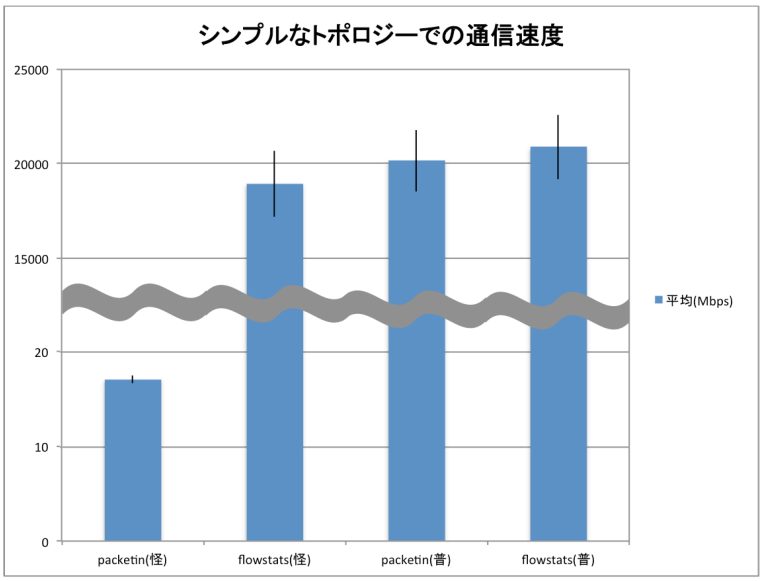

図 4.7　実験 1 の結果

出ていない．方式 A で怪しい通信先に通信するのは，その他の方式の 0.1%程度の速度となってしまうため，方式 A は実用性に欠ける．

4.3.4　実験 2: 怪しい通信先数の増加に伴うスループットの変化

実験の目的と内容

監視対象となる怪しい通信先を増やした時のスループットの変動について調査した．実験 4.3.3 において方式 A は実用的でないと判断したため，方式 B に対して実験を行った．実験 4.3.3 と同様の構成で，怪しい通信先として監視する対象の数を変数とした (図 4.8)．各

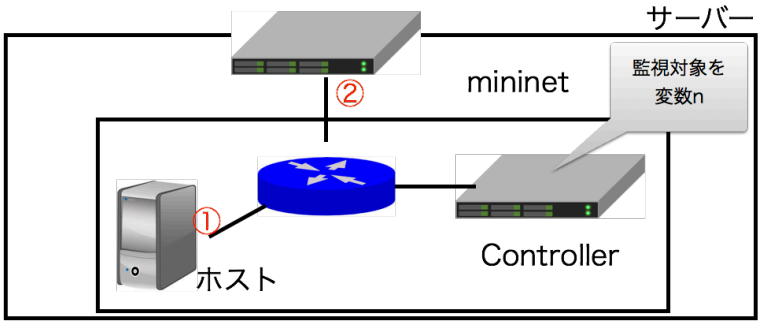

図 4.8　実験 2 の概要図

方式において監視対象を 10, 100, 1000 と変化させ，そのうちの 1 つに対して通信する際のスループットを計測した．

実験結果と考察

図 4.9 に監視対象を増やした時のスループットを 10 回計測した平均を示す．なお，グラフのエラーバーは標準偏差を利用した．グラフより，方式 B は監視対象が増えても通信速度に影響がないことが判明した．方式 B では，定期的に Controller で Stats Request が発行され，Switch は統計情報を返すため，ネットワーク制御とログの収集と解析が完全に分離されているわけではない．しかしながら，監視対象が増えても通信速度に影響が見られないため，OpenFlow による Stats 管理はフローテーブルの増加に強く実装されていることがわかる．

4.3.5　実験 3: Switch にひもづく Host 数と通信量増加に伴うスループットの変化

実験の目的と内容

Switch にひもづく Host 数が増え，Switch に流れるパケットが増えた時の怪しい通信先へのスループットを調査した．方式 B と方式 B において Flow Stats を要求しない方式に対して実験を行った．Switch1 台，Controller1 台に対し，Host を n 台接続したトポロジーのネットワークを構成し，怪しい宛先を 1 つ監視対象とした (図 4.10)．Switch に流れるパケットを増やすため，スループットを計測する Host 以外の Host から隣のホストに対し，*ping* を飛ばした．なお，随時パケットを流すために $-f$ オプションをつけ，ある程度のサイズにな

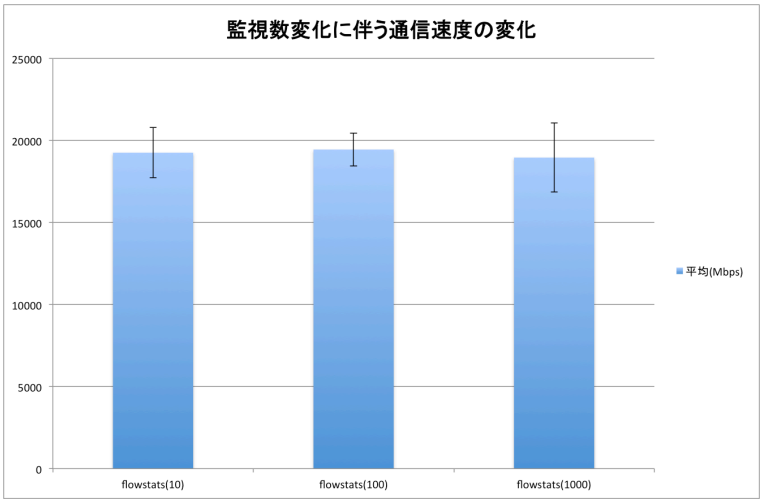

図 4.9　実験 2 の結果

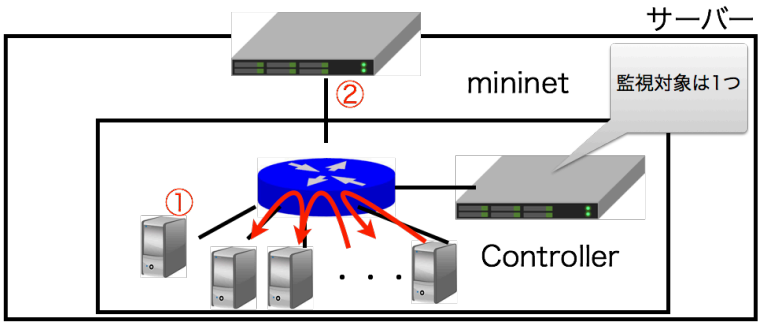

図 4.10　実験 3 の概要図

るよう 1 パケットの最大値を取るようにサイズを調整した結果，$ping - f - s1492 < IP$ アドレス $>$ を実行した．各方式において，Switch にひもづく Host を 10, 20, 50 台と変化させ，怪しい通信先への通信のスループットを計測した．

実験結果と考察

図 4.11 に Switch にひもづく Host 数と通信量増加が増やした時のスループットを 10 回計測した平均を示す．左半分が方式 B で実験したグラフで，右半分は方式 B において Flow Stats Request を飛ばさないで実験を行なったグラフである．なお，グラフのエラーバーは標準偏差を利用した．グラフより，方式 B は Switch に入るパケットが増加すると通信

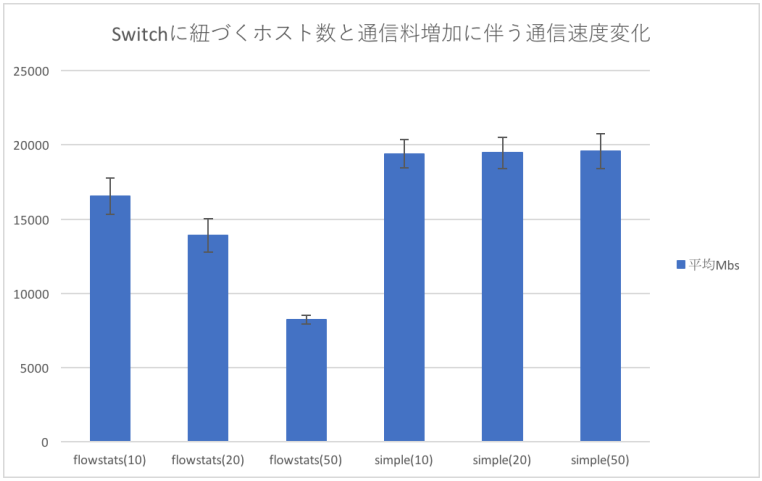

図 4.11　実験 3 の結果

速度に影響を受けることがわかる．Flow Stats の要求をしていない時はスループットの低下が見られないので，Flow Stats の処理が遅いと考えられる．実験 4.3.3 で変化がないのは，Switch の負荷が少なかったため Flow Stats の遅延が出なかったと考えられ，実験 4.3.4 の結果は，フローテーブル一つ一つではなくまとめて処理するように実装されているからだと考えられる．しかしながら，Flow Stats の処理が通常のパケット転送より遅いため，Switch へのトラフィックがある程度増えると，Switch が処理できるスループットを超えてしまい処理が遅延して通信速度が落ちる．

4.3.6　OpenFlow での実装についての考察

本実験から方式 A はシンプルな環境でも怪しい通信先へのスループットが 0.1%まで低下することがわかった．これは，怪しい通信先への通信全てを Packet In させていることが原因のため，方式 A は Controller に負荷をかけすぎるという問題があることがわかる．そのため，現実の複雑なネットワークに導入した時にネットワークが機能しない危険性がある．また，怪しい通信先への通信の速度低下は通常のネットワーク利用の妨げにつながるため不適切である．さらに，ログ管理のために Controller を経由するというのは実装として不適切ではないかとも考えられる．以上のことから，方式 A で提案システムを実装するのは不適切だと考えられる．

方式 B では方式 A のようにスループットが極端に下がることはなかった．しかし，実験 3 より方式 B は Switch の負荷が高まった際にスループットが減少する．20 台が常に通信

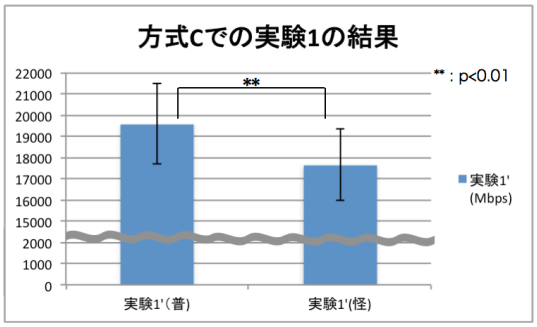

図 4.12　実験 1’ の結果

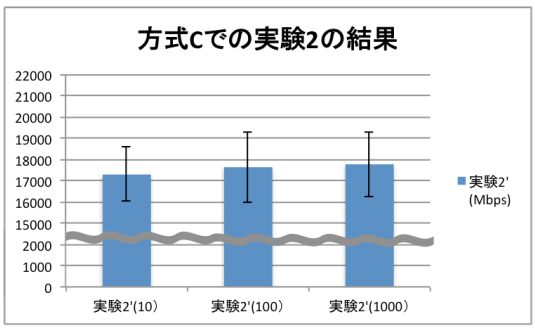

図 4.13　実験 2’ の結果

している状況で 33%程度スループットが減少し，さらに Switch の混雑が増えるとその分減少する．また，方式 B で利用している Flow Stats ではパケットの回数とサイズの統計情報しか扱えないため，厳密にトラフィックログを管理しているわけではない．ゆえに，時間などの要素を利用できずトラフィックログとして不十分である．したがって，方式 B で実装するとが適切とは言い切れない．

以上のことから，方式 AB 共に実装は可能だが実用的でないと言える．

4.4　OpenFlow を拡張した実装についての検討

4.3 節の実験により，提案システムの実装に対して方式 AB は不適切と判明した．そこで，3.4.2 項で提案した OpenFlow を拡張してトラフィックログの管理・解析とネットワークの制御を分離した手法 (以下，方式 C) のプロトタイプを作成し，実験 4.3 で行った 3 つの実験でスループットの調査をした．

4.4.1　実験 1’: 最もシンプルな条件でのスループット

実験結果と考察

図 4.12 に各プロトタイプにおいて怪しい通信先に通信した場合と普通の通信先に通信した場合のスループットを 50 回計測した平均を示す．なお，グラフのエラーバーは標準偏差を利用した．実験結果に対し t 検定を行なったところ，怪しい通信先との通信と普通の通信ではスループットに有意差が出た．そして，平均では怪しい通信先との通信が約 10%スループットが減少している．

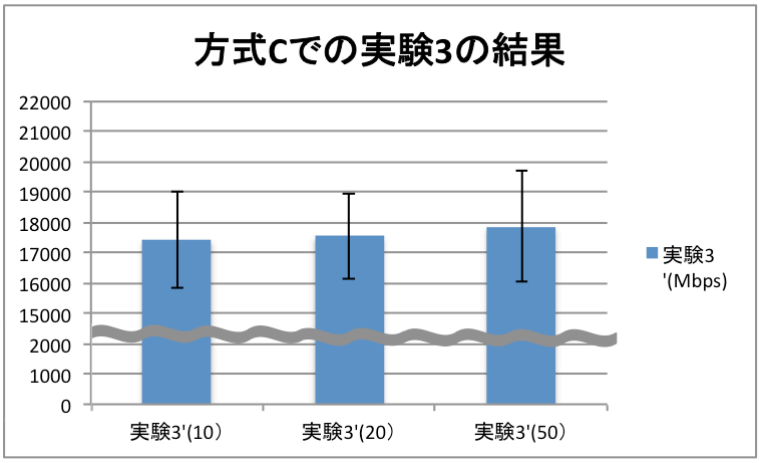

図 4.14　実験 3’ の結果

4.4.2　実験 2’：怪しい通信先数の増加に伴うスループットの変化

実験結果と考察

図 4.13 に監視対象を増やした時のスループットを 50 回計測した平均を示す．なお，グラフのエラーバーは標準偏差を利用した．実験結果に，分散分析を行なったところ有意差は見られなかった．

4.4.3　実験 3’：Switch にひもづく Host 数と通信量増加に伴うスループットの変化

実験結果と考察

図 4.14 に Switch にひもづく Host 数と通信量増加が増やした時のスループットを 50 回計測した平均を示す．なお，グラフのエラーバーは標準偏差を利用した．実験結果に，分散分析を行なったところ有意差は見られなかった．

4.4.4　OpenFlow を拡張した実装についての考察

本実験の結果より，OpenFlow を拡張した実装 (方式 C) は監視対象の怪しい通信先の数や Switch の混雑状況などの環境による速度低下の影響は少ないと考えられる．ただし，普通の通信先に比べ 10%程度スループットが減少するが，怪しい通信先への通信であることを考慮すると許容範囲内だと考えられる．したがって，方式 AB に比べ速度低下の課題が

少ないと考えられる．また，方式 C では方式 B と違いトラフィックログの情報が多いため，セキュリティインシデント対応やトラフィックログから脅威を検知する手法と組み合わせることができる．しかしながら，トラフィックログの管理と解析がネットワーク制御から分離されるため，通信フローを止めるまでに一定の時間が必要となる．したがって，解析対象のパケットそのものを止めることができなく，解析してから通信を止めるまでかかる遅延の時間分だけ流出の危険性がある．ただし，遅延を考慮した脅威判定とすることで影響を抑えることができると考えられる．

第5章　考察

本章では，提案システムの出口対策としての有効性と提案システムを実現する実装方法に関する考察と提案システムの課題について考察する．

5.1　実験結果について

5.1.1　提案システムの情報流出を抑制する出口対策としての有効性について

本節では予備検討と提案システムの有効性検討の結果をもとに提案システムの情報流出を抑制する出口対策としての有効性について議論する．

SDN で実現する利点について

4.1.1 項の結果から，SDN を用いた出口対策を実現することで速さ，簡単さ，柔軟性において従来のネットワークシステムより優れた対策が可能なことが判明した．ソフトウェア制御のため，ネットワーク経路をプログラマブルに変更できるようになり，人を介さずに対応できることから，素早い対応が可能でとなる．細かいネットワーク設定が不要でコマンドを打つだけ，またはプログラムが自動で対応するので，簡単なオペレーションで操作できる．そして，ネットワーク経路をプログラマブルに変更できることによって，特定通信の遮断や怪しい通信のみ監視を強化するなど柔軟な対策が可能になる．これらの機能は，ハードウェアによる転送管理が行われる従来のネットワークシステムでは実現は困難である．したがって，3.1 節で述べた 3 つの要件 (素早さ簡単さ柔軟性) を満たすので，SDN でネットワークを構成するのは有効だと考えられる．

情報流出の抑制について

4.2 節の結果から，提案した脅威度判定のアルゴリズムは大規模なデータを持つシステムに対しては効果がある可能性があることがわかった．例えば，式 3.6 の左辺の値を 1000 とし，通信サイズ制限を 1200 バイトとした場合，従来の対策では攻撃を受けた場合，数百万

人という極めて多くの情報が流出した危険性があったが，提案システムを導入すれば，1 万人程度の流出に抑えることができる．式 3.6 の左辺は，1 台のホストが STIX で共有された怪しい通信先リストの中で通信している通信先数とその回数の総和の掛け算である．4.1.2 項の結果から STIX には平均 13.9 個の怪しい通信先があるので，仮にすべての怪しい通信先に通信していた場合は，通信回数の総和の上限が 70 回程度で先述の 1 万人程度の抑制ができる．つまり，怪しい通信先の中で一つだけと通信しているときは 1000 回通信が可能で，すべての怪しい通信先と通信しているときは 70 回程度通信が可能である．怪しい通信先全てと通信しているときは，危険性が高いので 70 回通信を許可するのは危険な可能性が高い．そのため，制限を強くし，1 万人より流出人数を抑制することも可能だと考えられる．以上のことから，数千から 1 万人よりも十分に大きなデータを保持している組織にとっては，流出抑制できる可能性があると言える．

しかしながら，守るべき情報が全て流出してしまう前に措置をとることが保証されないため要件 2 を満たしているとは言えない．より早く判定を下せるアルゴリズムにするか，守るべき情報のサイズをパラメーターとして設定できるアルゴリズムを検討する必要がある．

提案したアルゴリズムで使用している脅威指標が直感にそぐわない点も問題である．通信回数が少ない間は脅威指標は少なく，途中から急に数値が増加し，収束するという挙動が適切だと考えられる．このことを踏まえたアルゴリズムの検討が必要である．

5.1.2　提案システムを実現する実装方法について

4.4 節の実験結果から，脅威判定の結果を元に通信遮断や認証をするシステムを実現するには方式 C(Switch のログを使えるよう OpenFlow を拡張する) が適切だと考えられる．サイバー攻撃のセキュリティインシデントに対応する際にトラフィックログを解析するのは重要である．そのため，トラフィックログを収集しかつ即座に確認可能な状態にするのは大事である．また，トラフィックログを収集し SDN と連携可能とすれば，トラフィックログから APT を検知する手法 [42][3] と組み合わせることが可能になる．これらのことを踏まえると，方式 B(Flow Stats を利用する) は情報が統計データとして丸められてしまうため，不適切だと考えられる．本研究では，安全性と利便性のトレードオフを考慮し，通常のネットワーク利用を可能な限り阻害しないことを目的の 1 つとしている．そのため，怪しい通信先への通信が極端に遅くなる方式 A(Packet In を駆使する) も不適切だと考えられる．したがって，トラフィックログとしての情報を損なわず，通信速度の影響もない方式 C が提案システムの実現方法に適していると言える．また，方式 C は方式 A や B と違

い Contorller からログ収集と解析機能が分離されているため，実装に適している．機能が分離されることにより，図 3.5 のようにシンプルな構成となる．シンプルな構成になることで，実装が簡単になる．さらに，機能が分離されているため，問題が生じた際に問題の発見が容易になる．つまり，ネットワーク制御の機能とログ収集と解析機能が分離するのは，実用性の面でも適切であり，簡単さを満たしていると言える．

5.2　提案システムの課題について

5.2.1　Switch のログを使えるよう OpenFlow を拡張する場合の課題

方式 C は既存の OpenFlow ではトラフィックログを収集する機能がないため，既存の OpenFlow では実現できない．同様に，P4[43] という新しい SDN プロトコルも Control Plane と Data Plane を分離することを主目的としているため，トラフィックログを処理する機能はない．したがって，OpenFlow または P4 を拡張してトラフィックログの収集機能を追加する必要がある．本論文では，tcpdump を利用して同様の機能を実現したが，この手法は Switch が Linux として動いていることを要求する．さらに，Network Interface Card(NIC) の数だけ tcpdump するプロセスが走るため，NIC 数が多いと負荷が高くなる危険性がある．そのため，Switch に流れるパケットのログを Linux OS を介さない Switch の機能として，まとめて処理できるように設計するべきだと言える．この際，通常の転送処理と同一のパイプラインに含まれるとトラフィックの増加に伴い遅延する危険性があるため，通常の転送処理とトラフィックログの処理は別々のパイプラインに分離する必要があると考えられる．

方式 C では，全ての通信に対して解析をするのではなく定期処理によって解析をするため結果を反映するまで一定の時間を要する．そのため，この対応までにかかる遅延の分だけ情報流出の危険性が高まると考えられる．ゆえに，定期処理の間隔によってどの程度危険性が高まるのかや，脅威度の増加予測と組み合わせるなどの工夫を検討する必要がある．例えば，定期処理 (バッチ処理) の間隔を t_b，解析にかかる時間を t_a，Controller に通信するのにかかる時間を t_c, Controller から Switch に反映するまでにかかる時間を t_s とすると，脅威判定の結果が閾値を超える通信が流れてから対応が完了するまでの時間 (T) は，最大で定期処理の時間間隔分を待ったときなので下記の式を満たす．

$$T \leq t_b + t_a + t_c + t_s \tag{5.1}$$

総務省の「通信自由化以降の通信政策の評価と ICT 社会の未来像等に関する調査研究」([**?**])

によると家庭向け固定通信の通信速度は 2000Mbps を超えている．つまり，1 秒間に 250M バイト情報が流出する恐れがあり，これは個人情報として換算すると 220 万人程度の情報となる．よって，回線速度が早いと遅延の影響によって多大な流出が起こる危険性がある．したがって，遅延を抑えるまたは，遅延の影響を少なくする必要がある．遅延は式 5.1 の条件を満たすが，右辺の要素を小さくするのは限界がある．よって，遅延の影響を少なくすることを検討するのが重要だと考えられる．遅延の影響を少なくする方法は，脅威判定が下ってから遅延することを考慮して閾値を下げることが挙げられる．一定値閾値を下げる手法や，脅威度の増加具合から遅延時間 (T) 秒後の脅威度を予測する手法などが考えられる．

第6章　結論

6.1　結論

本研究では，脅威情報を自動解析しアクセス制御することで標的型攻撃による情報流出を抑制するシステムを構築することを目的とした．このシステムを実現するために，SDN によって脅威情報を自動解析し，怪しい通信先との通信を監視し状況に合わせて通信遮断やアクセス確認などの対応をとるシステムを提案した．本論文では，提案システムの実現に向けて脅威判定のアルゴリズムや SDN の実装方法について検討した．まず，予備検討として SDN によって内部ネットワークを構築する有効性を検討するために，素早さと簡単さ，柔軟性の 3 つの観点において従来のネットワークシステムでの出口対策の場合と定性的に比較した．議論の結果，SDN による出口対策が 3 つの観点全てにおいて優れた対策であることが示せた．素早く簡単に対応できることによって，出口対策として流出までに対応する可能性が向上する．SDN のプログラマブルな性質による柔軟性は，怪しい通信のみ監視を強化するなどの効果的な対策を実現可能とする．次に，STIX で共有された怪しい通信先に対して通信サイズ制限や通信回数による脅威判定を行うことで個人情報の流出に対しては数万人程度の流出に抑えられるという結果を得られた．この結果から，日本年金機構や JTB などの個人情報流出事件などのような大規模なデータベースを持つシステムにおいては，情報流出抑制の可能性があると言える．しかし，守る対象の情報が全て流出する前に対処できる保証がないため，提案した通信回数による脅威判定のアルゴリズムだけでは不十分だと考えられる．提案した脅威判定のアルゴリズムについて検討する一方で，提案システムを実現する実装方法について，Packet In を駆使する方法と Flow Stats を利用する方法，Switch のログを使えるよう OpenFlow を拡張するの 3 方式についてスループットの調査をした．スループットの低下は通常の利用に影響することを考慮した結果，Switch のログを使えるよう OpenFlow を拡張する方式が良いことがわかった．さらに，セキュリティインシデント対応時にトラフィックログが必要なことやネットワーク制御機能とログ機能が分離することによって実装が易化することから，実用性の面でもこの方式が適していることも議論した．

6.2　今後の課題

今後の課題としては，要件 2 を満たす脅威判定アルゴリズムの再検討や OpenFlow の拡張，より複雑な実際のネットワークでの実装と性能検証があげられる．要件 2 を満たすために，従来のトラフィック情報を用いた脅威検知の研究などを参考にしてより早く脅威判定を下せるアルゴリズムにするか，守るべき情報のサイズをパラメーターとして設定できるアルゴリズムを検討する必要がある．OpenFlow を拡張し Switch のログを利用する機能実現することによって，提案システムを完成させることができる．この時，5.2 で述べた課題を考慮して，遅延を考慮した脅威判定システムを検討する必要がある．また，本論文の実験では，性能への影響の原因特定の簡単化のためにシンプルなネットワーク構成で実験を行なった．そのため，実際にネットワークを組む場合に実用に耐えうる性能かの検証が必要である．同時に，脅威判定の閾値を決めるための指標として，通信サイズ制限がどの程度通常のネットワーク利用に影響するかの検証も必要である．可能な限り通常のネットワーク利用を阻害させないことを目的としているため，ユーザーがストレスなく利用できるのが望ましい．

謝辞

本研究を進めるにあたり，修士課程 2 年間に渡って研究や発表，論文執筆などご指導していただいた指導教官である関谷勇司准教授に深く感謝いたします．また，CNL ミーティングにおいて研究の進め方，研究の方向性など多くの意見指導してくださった工藤知宏教授，中山雅哉准教授，佐藤周行准教授，小川剛史准教授，中村文隆助教，妙中雄三助教，宮本大輔助教に心より感謝します．JPCERT/CC の満永拓邦様には，セキュリティ分野について現場の様子も踏まえ詳細に教えていただきました．満永拓邦様のご助力がなければ，研究をここまで進めることはできませんでした．深く感謝いたします．研究室の先輩，同期，後輩の皆様には，研究についての相談や雑談などに付き合っていただき，大変充実した修士課程を過ごすことができました．最後に，浪人や休学など文句を言わず自由に過ごさせてくださった両親には心から感謝します．この場を借りて，皆様に厚くお礼申し上げます．

発表文献

国内会議 (査読なし)

1. 佐藤 康次, 関谷 勇司. “SDN を用いた Network 監視によるデータ漏えい防止機構の検討”, 信学技報, vol. 115, no. 484, IN2015-139, pp. 183-188 (2016 年 3 月).

2. 佐藤 康次, 関谷 勇司. “出口対策に向けた耐感染性を有したネットワーク監視並びに防御システムの検討”, 信学技報, vol. 116, no. 361, IN2016-82, pp. 91-96 (2016 年 12 月).

参考文献

[1] 警視庁. 平成 27 年上半期のサイバー空間をめぐる脅威の情勢について. https://www.npa.go.jp/kanbou/cybersecurity/H27_kami_jousei.pdf, 2015.

[2] Nist sp800-39: Managing information security risk:organization, mission, and information system view. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-39.pdf.

[3] Ross Brewer. Analysis of high volumes of network traffic for advanced persistent threat detection. *Network Security*, Vol. 2014, No. 4, pp. 5–9, 2014.

[4] P. Hu, H. Li, H. Fu, D. Cansever, and P. Mohapatra. Dynamic defense strategy against advanced persistent threat with insiders. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 747–755, April 2015.

[5] Inkyung Jeun, Youngsook Lee, and Dongho Won. *A Practical Study on Advanced Persistent Threats*, pp. 144–152. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[6] 不正アクセスによる個人情報流出の可能性について. http://www.jtbcorp.jp/jp/160614.html, 2016.

[7] Michele Colajanni Alessandro Guido Mirco Marchetti, Fabio Pierazzi. Advanced persistent threats: minimising the damage. *Computer Networks*, Vol. 109, No. 2, pp. 127–141, 2016.

[8] Daesung Moon, Hyungjin Im, Jae Dong Lee, and Jong Hyuk Park. Mlds: Multi-layer defense system for preventing advanced persistent threats. *Symmetry*, Vol. 6, No. 4, pp. 997–1010, 2014.

[9] 橋本賢一郎, 遠峰隆史, 関谷勇司. 昨今のサイバー攻撃の手法とその対策について : Interop tokyo 2014 shownet における結果からの考察 (ネットワーク研究開発テスト

ベッド運用・利用, 一般). 電子情報通信学会技術研究報告. IA, インターネットアーキテクチャ, Vol. 114, No. 236, pp. 51–56, sep 2014.

[10] 経済産業省 独立行政法人情報処理推進機構. サイバーセキュリティ経営ガイドライン ver 1.0. http://www.meti.go.jp/press/2015/12/20151228002/20151228002-2.pdf, 2015.

[11] 独立行政法人情報処理推進機構 技術本部セキュリティセンター. 標的型攻撃／新しいタイプの攻撃の実態と対策. https://www.ipa.go.jp/files/000024542.pdf, 2011.

[12] 独立行政法人情報処理推進機構 技術本部セキュリティセンター. 「新しいタイプの攻撃」の対策に向けた設計・運用ガイド. https://www.ipa.go.jp/files/000017308.pdf, 2011.

[13] 総務省. 標的型攻撃への対策 | 情報管理担当者の情報セキュリティ対策 | 企業・組織の対策 | 国民のための情報セキュリティサイト. http://www.soumu.go.jp/main_sosiki/joho_tsusin/security/business/admin/07.html.

[14] X. Wang, K. Zheng, X. Niu, B. Wu, and C. Wu. Detection of command and control in advanced persistent threat based on independent access. In *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2016.

[15] 国内標的型サイバー攻撃分析レポート 2015 年版. https://app.trendmicro.co.jp/doc_dl/select.asp?type=1&cid=161, 2015.

[16] H. Balinsky, D. S. Perez, and S. J. Simske. System call interception framework for data leak prevention. In *2011 IEEE 15th International Enterprise Distributed Object Computing Conference*, pp. 139–148, Aug 2011.

[17] David (Yu) Zhu, Jaeyeon Jung, Dawn Song, Tadayoshi Kohno, and David Wetherall. Tainteraser: Protecting sensitive data leaks using application-level taint tracking. *SIGOPS Oper. Syst. Rev.*, Vol. 45, No. 1, pp. 142–154, February 2011.

[18] JPCERT コーディネーションセンター. ”サイバー攻撃の高度な解析と情報共有のあり方について” jpcert/cc に置ける取り組み ～これまでとこれから～. https://www.ipa.go.jp/files/000027401.pdf, 2013.

[19] E. Asgarli and E. Burger. Semantic ontologies for cyber threat sharing standards. In *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, pp. 1–6, May 2016.

[20] STIX - Structured Threat Information eXpression. http://stixproject.github.io/.

[21] 脅威情報構造化記述形式 stix 概説. http://www.ipa.go.jp/security/vuln/STIX.html, 7 月 2015 年.

[22] 経済産業省. IT 人材の最新動向と将来推計に関する調査結果を取りまとめました. http://www.meti.go.jp/press/2016/06/20160610002/20160610002.pdf, 2016.

[23] ＪＴＢ個人情報７９３万件流出か？...標的型攻撃の巧妙な手口： 科学： 読売新聞（YOMIURI ONLINE）. http://www.yomiuri.co.jp/science/goshinjyutsu/20160615-OYT8T50004.html, 2016.

[24] Chia-Mei Chen and Hsiao-Chung Lin. Detecting botnet by anomalous traffic. *Journal of Information Security and Applications*, Vol. 21, pp. 42 – 51, 2015.

[25] M. N. Sakib and C. T. Huang. Using anomaly detection based techniques to detect http-based botnet c c traffic. In *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2016.

[26] Changhoon Yoon, Taejune Park, Seungsoo Lee, Heedo Kang, Seungwon Shin, and Zonghua Zhang. Enabling security functions with sdn: A feasibility study. *Computer Networks*, Vol. 85, pp. 19 – 35, 2015.

[27] Floodlight OpenFlow Controller -Project Floodlight. http://www.projectfloodlight.org/floodlight/.

[28] C. Cho, J. Lee, E. D. Kim, and J. d. Ryoo. A sophisticated packet forwarding scheme with deep packet inspection in an openflow switch. In *2016 International Conference on Software Networking (ICSN)*, pp. 1–5, May 2016.

[29] 電子政府ガイドライン作成検討会セキュリティ分科会. 電子政府ガイドライン作成検討会 セキュリティ分科会報告書. http://www.kantei.go.jp/jp/singi/it2/guide/security_guide_line/siryou2.pdf, 2010.

[30] 菊池浩明. サイバーセキュリティ分野における研究開発への期待, nict サイバーセキュリティシンポジウム 2016. http://www2.nict.go.jp/csri/plan/H28-symposium/pdf/kikuchi.pdf, 2016.

[31] R. Math and A. Goje. Impact of iot (internet of things) on growth of cloud applications in service sector with specific reference to western maharashtra. In *2016 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, pp. 153–156, Oct 2016.

[32] Martin Casado et al. Abstractions for Software-Defined Networks. http://cacm.acm.org/magazines/2014/10/178789-abstractions-for-software-defined-networks/fulltext.

[33] Open Networking Foundation (ONF). https://www.opennetworking.org/sdn-resources/sdn-definition.

[34] OpenFlow - Open Networking Foundation. https://www.opennetworking.org/sdn-resources/openflow.

[35] Open vSwitch. http://openvswitch.org/.

[36] MongoDB Documentation. https://docs.mongodb.com/.

[37] 梅田三千雄. 日本の苗字の計量的分析. 情報処理学会論文誌, Vol. 40, No. 3, pp. 796–804, Mar. 1999.

[38] 明治安田生命. 明治安田生命名前ランキング. http://www.meijiyasuda.co.jp/enjoy/ranking/year_men/boy.html, 2016.

[39] 日本郵政. 郵便番号データダウンロード. http://www.post.japanpost.jp/zipcode/dl/kogaki-zip.html, 2016.

[40] nuttcp. https://www.nuttcp.net/Welcome Page.html.

[41] Ryu SDN Framework. http://osrg.github.io/ryu/.

[42] K. F. Hong, C. C. Chen, Y. T. Chiu, and K. S. Chou. Scalable command and control detection in log data through uf-icf analysis. In *2015 International Carnahan Conference on Security Technology (ICCST)*, pp. 293–298, Sept 2015.

[43] P4: high-level language for programming protocol-independent packet processors. http://onrc.stanford.edu/p4.html.