Vol.2014-CSEC-66 No.18
Vol.2014-SPT-10 No.18
2014/7/3

SDN によるマルウェア調査のためのネットワーク切り替え手法

来間一郎[†1]　甲斐賢[†1]　木城武康[†2]　磯部義明[†1]

マルウェア感染への対処には，感染端末をネットワークから隔離することと，マルウェア活動による被害状況を調査することが必要となる．しかし，隔離されたことを検知し挙動を変化させるマルウェアに対しては，隔離と被害状況の調査を両立するのが難しいという問題がある．本研究では，ＳＤＮ技術を活用し，通信経路を個別に適切なタイミングで切り替えることで，マルウェアに検知されずに感染端末を隔離するネットワーク切り替え手法を提案した．さらに，提案手法に基づいたプロトタイプによる基礎機能の評価を行い，必要条件を明らかにした．

Method for Network Switching to Support Investigation of Malware with SDN

ICHIRO KURIMA[†1]　SATOSHI KAI[†1]
TAKEYASU KISHIRO[†2]　YOSHIAKI ISOBE[†1]

On security incident response, isolation of malware infected client and investigation of the damage situation of malware activity are needed. However, it is difficult to carry out them at the same time, when the malware has function to detect change of network state and change activity. In this paper, we propose the method to isolate malware infected client avoiding being detected by malware by changing network timely and partly using SDN (Software Defined Networking). We evaluated basic functions of prototype and found one required condition.

1. はじめに

1.1 インシデント対処とマルウェアの動向

近年，標的型攻撃に代表されるサイバー攻撃は高度化・組織化が進んでおり，セキュリティインシデントレスポンスチーム(IRT: Incident Response Team)の重要性が増している．IRT には，攻撃の状況を把握し，何が起きたかを明らかにするとともに，被害拡大の防止や原因の根絶，攻撃を受けたシステムの復旧が求められる．

本研究では，組織内システムにおいて，LAN 内のクライアントがマルウェアに感染している可能性がある場合，システム管理者が当該クライアントの管理を IRT に引き渡すまでの状況を想定する．

IRT は，マルウェアの調査を行い，マルウェアの機能や行われた活動を把握する．必要に応じてマルウェアの疑いがあるファイルを検体として抽出し，隔離された実行環境で動作させて振る舞いを観察する場合もある．この目的は，

- マルウェアを特定し，駆除する
- マルウェアの活動による影響範囲を明らかにする
- 法的措置が必要になった場合の証拠を確保する

等である．この証拠として以下の情報収集が要件となる．

- マルウェア検体
- マルウェアの振る舞い
  - ファイルアクセス
  - レジストリアクセス
  - パケット通信

一方で，マルウェア感染が疑われるクライアントは，早急にネットワークから切り離すことが推奨される[1]．これには，

- ネットワーク上の他のホストに対して，攻撃や感染拡大を目的とした悪意ある通信を行うことを防ぐため
- 外部からの通信によって，クライアント上のマルウェアの活動に関する情報が変化することを防ぐため

等の理由がある．

これに対し，近年，置かれた環境によって活動を隠蔽・変更する機能をもつマルウェアが増えており，IRT による情報収集の障害となっている．

マルウェアが動作を決定する基準となる環境要因は，自身が動作しているクライアントの情報と，クライアントが接続しているネットワークの情報の２つに大別できる．

それぞれの情報別にマルウェアの動向を説明する．

1) クライアント情報

”Jerry.c”は，マルウェアの動的解析に使われることが多い仮想環境構築ソフトである VMware 製品の特徴である，ホスト OS・ゲスト OS 間での通信チャネルへリクエストを送信し，その応答によって自身が動作している環境が仮想環境上か否かを判別するコードである[2].

また，2012 年から 2013 年にかけて発見された”UpClicker”

†1 (株)日立製作所　横浜研究所
Hitachi, Ltd.
†2 (株)日立システムズ
Hitachi Systems, Ltd.

や”BaneChant”は，ユーザからの入力の有無に注目し，ユーザの通常利用時に見られる入力操作がない場合は動的解析にかけられていると判断して活動を制限するマルウェアである．

2013 年に発見されたトロイの木馬である”Nap”や”Hastani”は，起動してから活動を開始するまでの時間や活動可能な日程を設定しておくことで，動的解析中には活動せず，長期にわたって潜伏した後に活動するマルウェアである．

”Citadel”は，初期感染時にその環境特有の情報を自身に書き込むことで，後に検体として抽出され，他の環境で動的解析にかけられた場合は，検査環境の情報と自身が記録している環境情報とを比較して別環境であることを検知し，活動を停止するマルウェアである[3].

2) ネットワーク情報

CCC DATAset 2009[4]の，SHA1 ハッシュ値の先頭４桁が 393F，7190 の検体は，インターネットとの通信を完全に遮断した場合よりも，一部の通信を許可した場合の方が実行されるコード量が多い[5]．これは，クライアントに感染したマルウェアが，攻撃者が用意したサーバに接続し，攻撃者からの指示を受信したり，新たなマルウェアをダウンロードしたりすることで，攻撃を進めていくためである．この種のマルウェアを，インターネットから隔離された検査環境上で実行し，動的解析を試みた場合，動作に必要な通信ができないため，一部の機能が実行されず，本来の動作を調査できなくなる．より能動的な例としては，インターネットへの接続可否を確認し，接続不能であれば隔離された環境に置かれていると判断して動作を停止するマルウェアも確認されている[6].

また，LAN 内の他のホストに対する通信を定期的に試み，応答を監視することで隔離状況を検知するマルウェアの存在も示唆されている．マルウェアによっては，証拠隠滅だけでなく，感染クライアントのデータ全体を破壊する場合もあるため，ネットワークから切り離すことで感染端末内のデータを確保できない場合がある[7].

以上のように，近年は活動を隠蔽・変更する機能を持つマルウェアが増えており，インシデント対処における情報収集の障害となっている．

1.2 本研究の目的

本研究では，マルウェア感染が疑われるクライアントを発見してから，当該クライアントの管理をシステム管理者が IRT に引き渡すまでの支援において，以下を両立するため，有効なネットワーク切り替え手法を提案することを目的とする．

- 当該クライアントの通信をコントロールしてマルウェアをネットワークから隔離し，当該クライアントとネットワーク上の他のホストが互いに影響を及ぼさないようにする
- マルウェアが隔離されたことを検知できないように通信をコントロールし，マルウェアが挙動を隠蔽して後の調査に必要な情報が失われることを防止する

2. 従来技術

インターネットとの通信が必要なマルウェアに対しては，インターネットと通信可能な環境でマルウェアを動作させることは，更なる攻撃を許す可能性があるため，推奨されない．

青木らは，マルウェア検体を動的解析する場合に，サンドボックスからインターネットへの通信を監視し，C&C サーバとの通信や HTTP によるファイルダウンロードと判断できる通信のみ許可することで，安全を確保しながら動的解析可能なシステムを提案した[5].

セキュリティインシデント対処に SDN を利用する例としては，SDN 対応ネットワーク機器にセキュリティ機能を組み込んだ製品がある[8]．セキュリティスイッチは，レイヤ 2 のスイッチにセキュリティ機能を加えたもので，MAC アドレスによる認証や，悪性通信の検知により，通信を遮断する機能を備える．

3. 従来技術の課題

マルウェアがネットワーク状態の変化を検知する状況を分類すると以下の通りとなる．

- マルウェアの通信中にネットワーク状態が変化した場合，通信が中断されることから変化を検知する
- マルウェアが通信していない時にネットワーク状態が変化した場合，マルウェアが再度通信を開始した時，通信相手の応答が以前と違うことから変化を検知する

以上を踏まえ，

課題１． ネットワーク切り替え後に，マルウェアの通信先の応答が変化しないようにネットワークを切り替える

課題２． マルウェアの通信を中断しないようにネットワークを切り替える

の 2 点を課題と設定する．

4. 提案手法

4.1 SDN

近年の情報システムの変化として，ネットワーク仮想化への動きが挙げられる．普及が進むクラウドを中心に，サーバ，ストレージ，ネットワークなどのリソースを分離して管理・利用する技術の需要が高まっている．特に，サーバ，ストレージの仮想化技術の進歩と比較して遅れてきたネットワークの仮想化技術が近年になって急速に進歩している．

Vol.2014-CSEC-66 No.18
Vol.2014-SPT-10 No.18
2014/7/3

SDN を実現する代表的仕様である OpenFlow[a] ver1.3.2 には以下のような特徴がある[9].

(1) パケットの送受信を実行するスイッチと，スイッチを管理し送受信の規則を制御するコントローラに分かれたアーキテクチャを採用している．あらかじめ設定された規則に従いスイッチ単独で動作することも，パケットを処理するたびにコントローラに問い合わせすることも可能である．

(2) レイヤ 1 からレイヤ 4 までの情報について，パケットの識別に利用でき，また操作対象とすることが可能である．物理ポート番号，MAC アドレス，VLAN ID，IP アドレス，TCP/UDP ポート番号等でパケットを識別し，ヘッダ情報を書き換えて指定した物理ポートから出力させることができる．

以上のように，SDN は今後データセンタを初めとして多くの情報システムを支える基盤として組み込まれていくと期待されており，多くの状況で利用可能になると予想される．加えて，ネットワークの動的制御，詳細なルールの適用，マルウェアが干渉できないレイヤ 1 での操作が可能といった特徴から，マルウェアに検知されることなく通信経路を変更することに適している．

以上より，本研究の課題を解決するために SDN を活用することは有用といえる．

4.2 解決方針

提案手法では，前述の課題を以下のように解決する．

課題 1 に対しては，マルウェアの通信先によって選択的に経路を切り替えることで解決する．

通信経路を遮断するだけでは，以前に通信できていた通信先との通信ができなくなり，ネットワークの状態が変化したことを検知する手がかりとなる．これを防ぐために，ダミーサーバを用意し，マルウェアから本物のサーバへの通信をこちらに誘導する．ただし，LAN 内のサーバについては，同様のサービスを稼働させたダミーサーバを用意することは可能だが，インターネット上のサーバ，特に攻撃者が用意した C&C サーバについてはダミーを用意することは難しい．そこで，マルウェア感染クライアントからの通信を通信先サーバによって分類し，通信経路を変更可能なものについてのみ，通信経路をダミーサーバへ変更する方法をとる．

以下では，一例として，マルウェア感染クライアントからの通信を，LAN 内のファイルサーバに対するものについてはダミーサーバへ誘導し，プロキシサーバを介してインターネットへ通じるものについては変更しない状況を示す．

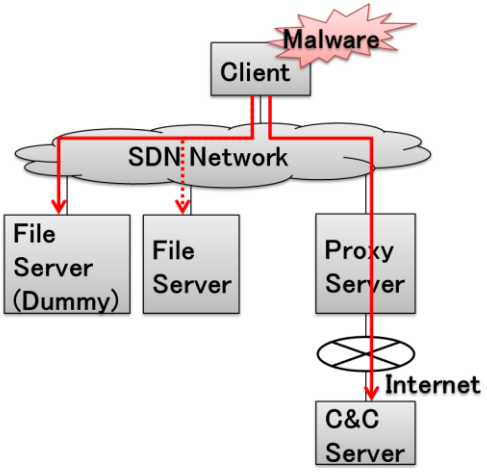

図 1　選択的経路切り替え
Figure 1　Selective Route Switching

課題２に対しては，以下２つのアプローチが考えられる．

- セッションと通信内容を維持したまま，マルウェア感染クライアントからの通信をダミーサーバへ誘導する
- マルウェア感染クライアントの通信状態を観測し，通信していないタイミングで通信をダミーサーバへ誘導する

前者はダミーサーバに機能を追加する必要があるため，本研究ではネットワーク部分のみで対応できる後者のアプローチを選択する．

さらに，マルウェア感染クライアントが通信していないタイミングでネットワーク切り替えを行ったとしても，切り替えとほぼ同時にマルウェアが通信を開始する場合，マルウェアの通信を中断させる可能性がある．これに対し，本報告では，統計的手法でマルウェア通信の開始タイミングを予測し，通信開始されないタイミングでネットワークを切り替える手法をとる．

4.3 タイミング予測手法

この手法は，ネットワーク切り替え時に通信を中断させないために，マルウェアの通信タイミングを予測し，通信開始可能性が低いタイミングを選んでネットワークを切り替える手法である．

以下では，単純化したネットワーク構成として以下のネットワーク構成を想定する．

a) OpenFlow は，Open Networking Foundation の登録商標です．

Vol.2014-CSEC-66 No.18
Vol.2014-SPT-10 No.18
2014/7/3

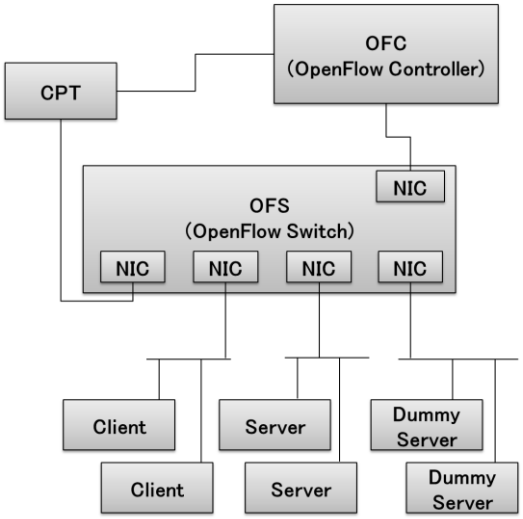

図 2　ネットワーク構成
Figure 2　Network Configuration

OpenFlow コントローラは，OpenFlow スイッチに接続され，スイッチの動作を制御する．OpenFlow スイッチはクライアント，サーバ，ダミーサーバをつなぎ，パケットを中継する．また，スイッチを通る，クライアント・サーバ間の通信パケットをミラーリングし，CPT に転送する．CPT は，取得したパケットから各クライアントの通信状況を把握し，OpenFlow コントローラに伝える．

4.3.1 タイミング予測原理

本手法では，ネットワーク上から観測できる情報で，マルウェア感染クライアントの通信タイミングを予測する．

マルウェアによる通信とその他の正常な通信は区別するのが難しいため，マルウェア感染クライアントからの全ての通信に対して処理を行う．

特徴量として利用するのは，通信ログから得られる過去の通信時間・非通信時間である．

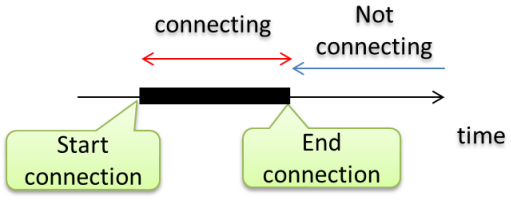

図 3　通信状態の表し方
Figure 3　Notation of Connection State

図 3 は，TCP 通信の様子を表している．時間軸上で，Start Connection において TCP 通信が開始され，End Connection の時点で終了する．この間の，TCP コネクションが張られている状態（Connecting）の継続時間を通信時間，それ以外の状態（Not Connecting）の継続時間を非通信時間と定義する．

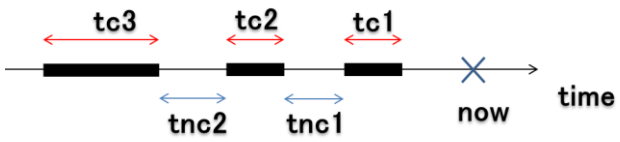

図 4　通信状態の時間変化
Figure 4　Translation of Connection State

あるクライアントから特定のサーバへの通信状態（TCP コネクションの状態）の時間変化は，図 4 のように表現できる．

この時，ネットワーク上を流れるパケットを取得することで，このクライアントの通信状態の変化を記録しておき，それを参照することで，過去の通信時間（tc(1), tc(2), tc(3)…）と非通信時間(tnc(1), tnc(2)…)を求めることができる．

ここで，一例として，非通信時間(tnc(x))とその直前の通信時間(tc(x+1))との間に関係があると仮定し，その関係を通信パターンとして，次の通信開始タイミングの予測に利用することを考える．組み合わせ(tnc(x), tc(x+1))の分布はクラスタリングによって複数の正規分布の重ね合わせで近似でき，重ね合わせの重み・平均ベクトル・共分散行列の値として表現される．これを通信パターンとする．近似方法としては，EM アルゴリズムなどの一般に知られている手法を利用する．

この例では非通信時間とその直前の通信時間の関係に着目したが，マルウェアの種類によって通信方式が違うため，次の通信が開始されるタイミングを予測する上で適切な着目要素も異なると考えられる．そこで，着目する要素の組み合わせを変えて通信パターンを複数算出し，分布がはっきり分かれるものほど予測に有用と考え，各分布のパラメタから有用度を定量評価して最適な通信パターンを選択する．

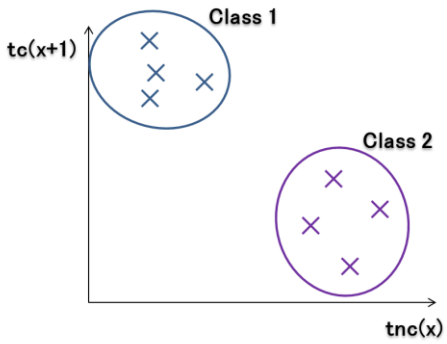

図 5　特徴量によるクラス分類
Figure 5　Classified Pairs of Feature Values

Vol.2014-CSEC-66 No.18
Vol.2014-SPT-10 No.18
2014/7/3

前述の通信パターンを用いた，(tnc(x), tc(x+1))を特徴量とした場合の，経路変更タイミングの決定方法について説明する．

通信経路変更対象クライアント・サーバの通信状態ログを取得し，取得済みの通信パターンと比較する．ログに記録された直前の通信時間 tc(1)の値と通信パターンの確率分布から，現在状態が各クラスに属する確率を求め，この確率で重み付けして各クラスの非通信時間の確率分布を重ね合わせ，現在の非通信状態が継続する時間の確率分布を得る．この分布から，確率が一定値未満の時間帯であれば経路変更可能と判断し，直前の通信終了時刻からの経過時間がその範囲にある時に経路変更を実行する．以上を図示したものが次の図である．

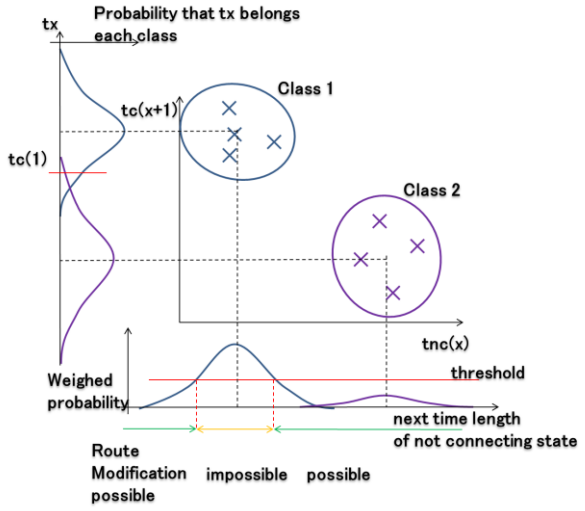

図 6　ネットワーク切り替えタイミング決定
Figure 6　Decision of Route Modification Timing

4.3.2 通信路の粒度

本手法では，マルウェア感染クライアントから，LAN 内のサーバ（インターネットに接続するためのプロキシサーバを除く）への通信路を監視し，通信していないタイミングでスイッチの設定を変更して，以降の通信をダミーサーバへ誘導する．制御する通信路の粒度としては以下が考えられる．

A) 送信元（クライアント）IP
B) 宛先（サーバ）IP
C) 送信元（クライアント）ポート番号

模式図を図 7 に示す．

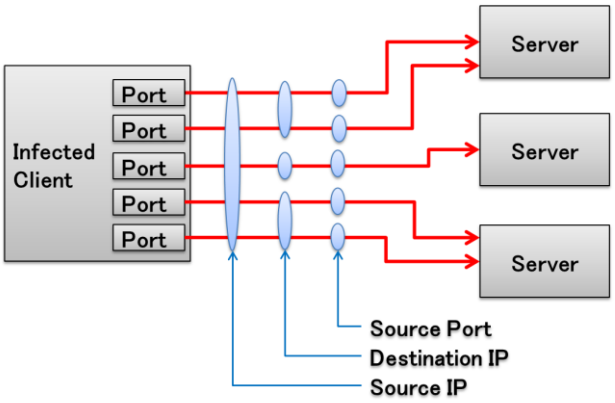

図 7　通信路の粒度
Figure 7　Granularity of Connection Routes

A) は最も多くの通信路をまとめて制御する方式，C) は通信路を細分化して制御する方式といえる．

A) は，スイッチの設定を変更するにあたり，少ない命令で全ての通信路を制御できるというメリットがある．しかし一方で，全ての通信路について，クライアントが通信を行っていないタイミングでしかネットワーク切り替えができず，切り替えのチャンスが来るまでに時間が掛かるという問題がある．

C) は，通信路を細かく制御できるため，クライアントが通信を行っていない通信路から順次切り替えが可能である．しかし，１クライアントにつき，最大で 65536 ポートの通信を監視して統計処理をする必要があり，スイッチの設定もポート単位で行うため，処理量が膨大になるという問題がある．

B) では，制御すべき通信路の数は LAN 内のサーバの数と等しく，扱う上で現実的な数である．また，通信先サーバごとに経路制御するため，切り替え可能になったサーバから順次ダミーサーバにつなぎかえ，安全を確保できる．

よって本手法では B) の粒度で通信路を扱う．

4.3.3 通信パターン取得タイミング

本手法では，切り替えタイミング算出のために，マルウェア感染クライアントから LAN 内のサーバへの通信を取得して蓄積し，統計処理にかける必要がある．ここで問題になるのは，いつ通信を取得するかである．

マルウェア感染が疑われるクライアントが発見されてから通信取得を開始すると，統計処理で十分な精度を得られる量の情報が蓄積されるまでに時間がかかり，ネットワーク切り替えの実行が遅れるという問題がある．

これを解決するためには，通常時から通信を取得して統計処理を施しておき，マルウェア感染が疑われるクライアントが発見された時点で，統計情報を利用できるようにしておく必要がある．

5. 評価

5.1 評価目的

前章までに検討した提案手法を実装する上での必要条件を検討するため，ネットワーク切り替えに要する時間を計測する．

OpenFlow コントローラがネットワーク切り替え指令を送信してから，実際に OpenFlow スイッチ上での処理が変化しパケットの流れが切り替わるまでには時間差がある．この間にマルウェアが通信を開始すると，ネットワーク切り替えが通信を阻害することになるため，マルウェアから環境変化として検知可能になる．提案手法では，この事態を防ぐため，マルウェアの通信開始タイミングを予測して回避するネットワーク切り替え方式をとっている．ネットワーク切り替えに要する時間は，タイミング予測時の必要条件に影響を与えると考えられるため，この値を計測し，必要条件を検討する．

5.2 評価環境

本評価では，OpenFlow コントローラ，OpenFlow 対応スイッチと，スイッチにつながるクライアント，サーバ，ダミーサーバからなり，コントローラからの指令によって各クライアント，サーバ間の通信経路を変更する機能を実装したプロトタイプを作成した．

評価環境は，ブレードサーバ上で，ハイパーバイザとして VMware ESXi を用いて複数の仮想マシンを作成し，それらがネットワークを構成するものとした．仮想環境上のネットワーク構成を以下の図に示す．

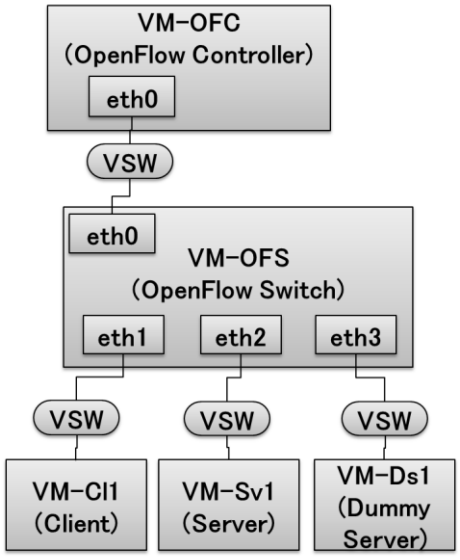

図 8　評価環境ネットワーク構成
Figure 8　Network Configuration of Evaluation Environment

また，本評価に使用した機器の構成は以下の表の通りである．

表 1　プラットフォーム
Table 1　Platform

Maker	Hitachi
Model	HA8000-bd/BD10
CPU	Intel Core(TM) i5 2.40GHz
Memory	8GB
Hyper Visor	VMware ESXi 5.0.0

表 2　仮想マシン設定
Table 2　VM configuration

VM-OFC	Allocated Memory	1GB
	OS	Ubuntu 12.10 32bit
	IP address	192.168.0.6
VM-OFS	Allocated Memory	IGB
	OS	Ubuntu 13.10 32bit
	IP address (eth0)	192.168.0.4
VM-Cl1	Allocated Memory	1GB
	OS	Ubuntu 13.10 32bit
	IP address	192.168.0.3
	MAC address	00:0c:29:bb:57:b8
VM-Sv1	Allocated Memory	1GB
	OS	Ubuntu 13.10 32bit
	IP address	192.168.0.5
	MAC address	00:0c:29:50:c8:11
VM-Ds1	Allocated Memory	IGB
	OS	Ubuntu 13.10 32bit
	IP address	192.168.0.5
	MAC address	00:0c:29:86:7c:c4

全ての仮想マシンには，OS として ubuntu をインストールした．VM-OFC は，仮想マシンに OpenFlow コントローラのフレームワークである Trema をインストールし，OpenFlow コントローラとして動作させた[10]．VM-OFS は，仮想マシンに Open vSwitch をインストールして，OpenFlow 対応スイッチとして動作させた．VM-Cl1 はクライアント，VM-Sv1 はサーバ，VM-Ds1 はダミーサーバにあたる．VM-Ds1 は VM-Sv1 に対応するダミーサーバとして，同一の IP アドレスを設定した．本構成では，ESXi 上の仮想スイッチ(VSW)は，各仮想マシンを接続する回線として利用し，ラーニングスイッチとしての機能は利用しなかった．そのため，無差別モードを「承諾」とし，リピータハブとして振る舞うよう設定した．

ネットワーク切り替えの前後で，スイッチに与える設定は以下の表の通りである．ネットワーク切り替え前は，スイッチの NIC eth1 と eth2 の間で相互にパケットを通過させる．ネットワーク切り替え後は，VM-Cl1 から VM-Sv1 へ

の IP パケットを，IP アドレスにより抽出して，宛先 MAC アドレスを VM-Ds1 のものに書き換えて eth3 から出力する．VM-Ds1 から VM-Cl1 への IP パケットは，送信元 MAC アドレスを VM-Sv1 のものに書き換えて eth1 から出力する．これにより， VM-Sv1 に通信しているのと同じ状態で VM-Cl1 から VM-Ds1 に通信させる．

表 3 ネットワーク切り替え前（通常時）スイッチ設定
Table 3 Switch Configuration (at normal time)

Matching Rules	Actins	Priority
In_Port: eth1	Send Out Port: eth2	0
In_Port: eth2	Send Out Port: eth1	0

表 4 ネットワーク切り替え後（IR 時）スイッチ設定
Table 4 Switch Configuration (at incident response)

Matching Rules	Actions	Priority
in_port: eth1	Send Out Port: eth2	0
in_Port: eth2	Send Out Port: eth1	0
In_Port: eth1 Dl_type: 0x0800 Nw_src: 192.168.0.3 Nw_dst: 192.168.0.5	Modify dl_dst: 00:0c:29:86:7c:c4 Send Out Port: eth3	10
In_Port: eth3 Dl_type: 0x0800 Nw_src: 192.168.0.5 Nw_dst: 192.168.0.3	Modify dl_src: 00:0c:29:50:c8:11 Send Out Port: eth1	10

5.3 ネットワーク切り替え時間計測

図 9 に計測時のパケットの流れを示す．

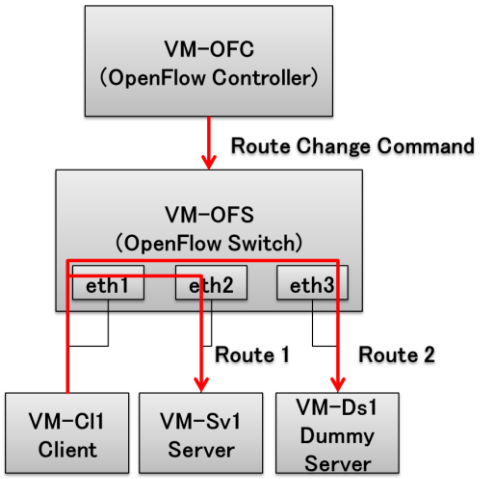

図 9 ネットワーク切り替え時間計測
Figure 9 Measuring Switch Configuration Setting Times

VM-OVC と VM-Cl1 の時刻を同期させ，VM-Cl1 から，VM-Sv1 および VM-Ds1 の IP アドレスに宛て，ペイロードに発信時刻を書き込んだ UDP パケットを連続して送出した．その後，VM-OVC から VM-OVS にネットワーク切り替え指令を送信し，VM-Cl1 から送信された IP パケットの出力先ポートを eth2，eth3 の間で変更することで，UDP パケットが届く先を VM-Sv1，VM-Ds1 の間で切り替えた．VM-Sv1，VM-Ds1 で受信したパケットのペイロードに書き込まれている発信時刻と，VM-OVC で切り替え指令を送信した時刻を比較し，切り替え指令送信から実際にスイッチ上の処理が変化するまでの時間を計測した（N=10）．
結果を以下に示す．

表 5 ネットワーク切り替え所要時間
Table 5 Switch Configuration Setting Times

Destination (Before)	Destination (After)	Average of Switching Time [ms]	Standard Deviation of Switching Time [ms]
VM-Sv1	VM-Ds1	1.45	2.27
VM-Ds1	VM-Sv1	1.95	1.17

ネットワーク切り替えに要する時間は，総合して 1.70±1.78[ms]となった．

6. 考察

ネットワーク切り替えと，マルウェアの通信開始がほぼ同時に開始された場合，ネットワーク切り替えのタイミングとしては以下 4 つが考えられる．

1．SYN パケットがスイッチを通過する前
2．SYN パケット通過後，SYN/ACK パケット通過前
3．SYN/ACK パケット通過後，ACK パケット通過前
4．ACK パケット通過後

図 10 にパケットの流れとタイミングの関係を示す．

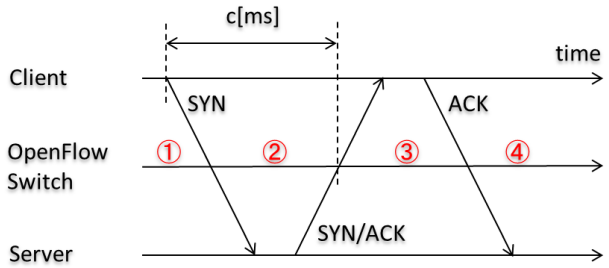

図 10 接続確立中のネットワーク切り替え
Figure 10 Switch Configuration Setting in Establish Connection

1 では，以降の通信がすべてクライアント・ダミーサーバ間で行われるため，問題は生じない．

2 では，SYN パケットはサーバに届くが，サーバからの SYN/ACK はクライアントに届かない．この時，クライアントが SYN 再送を行えば，以降の通信はダミーサーバへ届き，クライアント・ダミーサーバ間で接続が確立され，問題は生じない．

3 では，SYN/ACK はクライアントに届くが，クライアントからの ACK はダミーサーバへ届く．この時，ダミーサーバは RST を送り返し，接続を終了する．これは異常としてマルウェアから検知可能である．

4 では，クライアント・サーバ間の接続が確立した後，クライアントからの通信がダミーサーバへ誘導される．以後，クライアントからの通信が発生した場合，ダミーサーバは RST を送り返し，接続を終了すると考えられる．これは異常としてマルウェアから検知可能である．

クライアントの通信開始がコントローラに伝わり，切り替え可能か否かを判断するまでの時間を a[ms]，コントローラがネットワーク切り替え命令を出してからスイッチ上の処理が変化するまでの時間を b[ms]，クライアントの通信開始から上記タイミング 2 の終了までの時間を c[ms]とする（図 10）．a+b<c の時，ネットワーク切り替えと前後してマルウェア通信が開始されても，タイミング 2 以前に切り替えられるため，問題にはならない．それ以外の場合，予測される非通信時間が a+b-c[ms]より長い場合のみ切り替えが可能となる．

本評価環境において，b は 1.70±1.78[ms]であった．a は主にクライアントとコントローラ間の，c は主にクライアント，スイッチ，サーバ間の経路長によって影響を受けると考えられる．ネットワーク機器の配置を工夫することで，予測に求められる精度を調節することができる．

以上より，提案手法が必要となる条件，および有効となるために必要な条件を明らかにした．

7. おわりに

本研究では，マルウェア感染が疑われる端末を発見してから，当該端末をセキュリティインシデント対処チームに引き渡すまでの支援において，

- 当該クライアントの通信をコントロールしてマルウェアをネットワークから隔離し，当該クライアントとネットワーク上の他のホストが互いに影響を及ぼさないようにする
- マルウェアが隔離されたことを検知できないように通信をコントロールし，マルウェアが挙動を隠蔽して後の調査に必要な情報が失われることを防ぐ

を両立するため，SDN 技術を利用した有効なネットワーク切り替え手法を提案することを目的とした．

課題として

- 課題１．ネットワーク切り替え後に，マルウェアの通信先の応答が変化しないようにネットワークを切り替える
- 課題２．マルウェアの通信を中断しないようにネットワークを切り替える

を設定し，これらを解決する手法として，

- 特定経路切り替え
- マルウェア通信タイミング予測

からなるネットワーク切り替え手法を提案した．

提案手法について，プロトタイプを作成して評価を行い，ネットワーク切り替えに要する時間を 1.70±1.78[ms]と計測した．これを基に，ネットワーク構成および予測される非通信状態の継続時間から，ネットワーク切り替え可否を算出する方法を整理した．

以上より，ネットワーク状態の変化を検知して挙動を変化させるマルウェアに対しても環境変化を検知させずに通信経路を変更し，マルウェアの挙動の維持と，マルウェアの悪意ある通信の隔離を両立する上で，必要となる条件を明らかにした．

今後の課題として，ネットワーク状態を検知するマルウェア検体に対して動的解析を行い，本手法の有効性を検証する必要がある．

参考文献

1) コンピュータセキュリティインシデント対応ガイド, http://www.ipa.go.jp/files/000015367.pdf
2) Matthew, Carpenter. Tom Liston, Ed Skoudis.: Hiding Virtualization from Attackers and Malware, IEEE Security and Privacy, pp.62-65 (2007).
3) FFRI, Inc.: Citadel の解析から得られた近年のマルウェア傾向, 情報セキュリティ EXPO2014 春 (2014).
4) 畑田充弘, 仲津留勇, 寺田真敏, 篠田陽一: マルウェア対策のための研究用データセットとワークショップを通じた研究成果の共有, マルウェア対策研究人材育成ワークショップ (2009).
5) 青木一史, 川古谷裕平, 岩村誠, 伊藤光恭: 半透性仮想インターネットによるマルウェアの動的解析, 情報処理学会, コンピュータセキュリティシンポジウム論文集 (2009).
6) 長期潜伏，自らを削除--サンドボックスを回避する未知のマルウェア, http://japan.zdnet.com/security/sp_networksec/35047336/
7) マルウェアによるインシデントの防止と対処のためのガイド, pp.65 (2005). http://www.ipa.go.jp/files/000025349.pdf
8) HanDreamnet SubGate, http://www.handreamnet.jp/product/switch/
9) OpenFlow Switch Specification version 1.3.2, https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.2.pdf.
10) 高宮安仁, 鈴木一哉: OpenFlow 実践入門, pp.66 (2013).