動的解析結果に基づく共通的なマルウェア通信パターン抽出技術の検討

中田 健介† 青木 一史† 神谷 和憲† 角田 進† 大嶋 嘉人†

†NTT セキュアプラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11
{nakata.kensuke, aoki.kazufumi, kamiya.kazunori, kakuta.susumu,
oshima.yoshihito}@lab.ntt.co.jp

あらまし マルウェアは日々進化し続け，入口対策などで感染を完全に防ぐことは困難になりつつある．そこで，感染後の被害軽減を目的とした出口対策による感染の早期検知の重要性が増している．特に SIEM と呼ばれる様々なログを収集・分析する装置を活用し，マルウェアが行う不正な通信を早期に検知する感染端末検知技術が注目されている．感染端末の検知にはマルウェアを検知するための通信パターンを定義する必要があるが，検知精度と処理性能を両立させて多様なマルウェアを検知できる通信パターンを定義するのは難しい．本稿では，マルウェアの動的解析結果から得た通信ログに見られる共通的な通信パターンを，通信の特徴を捉えるルールの解析結果の時系列に着目して抽出する技術について提案する．また，マルウェアの通信ログおよび実網の通信ログに対して，提案手法で抽出した通信パターンを用いてマルウェア通信検知を行った．その結果，誤検知率を 0.5%以下に抑えつつ，従来手法と比較して検知率が 30%程度向上することを確認した．また，処理性能についても日次分析に適用可能であることを確認した．

A Method of Extracting Common Malware Communication Patterns from Dynamic Analysis Result

Kensuke Nakata† Kazufumi Aoki† Kazunori Kamiya† Susumu Kakuta†
Yoshihito Oshima†

†NTT Secure Platform Labolatories.
3-9-1 Midori-cho, Musashino-shi, Tokyo 180-8585, JAPAN
{nakata.kensuke, aoki.kazufumi, kamiya.kazunori, kakuta.susumu,
oshima.yoshihito}@lab.ntt.co.jp

Abstract

Since malware continues evolving, it is difficult to completely prevent infection. Post-infection countermeasure is important to reduce damage caused by infection. One of the promising approach is utilizing SIEM(Securty Information and Event Management) which analyzes various types of logs collected in networks and detects malware-infected hosts. In specifying malware-infected hosts through network log analysis, having communication patterns of malware with high true positive detection rate is necessary. Furthermore, the processing performance is important too. We propose a method of extracting common communication patterns of malware using dynamic analysis result. This method is focusing on event sequences which are triggered by predefined ruleset and effectively extracting common communication patterns as detection signature. Our evaluation results show that true positive rate is 30% superior to existing method while keeping false positive under 0.5%. The results also show that proposed method have a good processing performance enough to apply daily analysis.

1　はじめに

近年, 日々進化し続けるサイバー攻撃の脅威が増大している．サイバー攻撃とはコンピュータやインターネットなどを利用して, 攻撃対象のシステムやネットワークに不正に侵入し, 機密情報の詐取・破壊・改ざんを行ったり, 対象システムが提供するサービスを機能不全に追い込むものである．

サイバー攻撃の一つにユーザ端末に対する攻撃がある．ユーザ端末に対する攻撃ではコンピュータウィルスに代表されるマルウェアに感染させることを狙った攻撃が主流である．具体的にはマルウェアを仕込んだファイルを電子メールに添付して送信する手法がある．またはマルウェアを強制的にダウンロードさせられるサイトへの誘導を目的とした電子メールの送信が行われることも多い．

従来のサイバー攻撃では不特定多数に対して無差別に攻撃が行われる事が多かったが, ここ数年は特定の集団・組織・個人を標的とした標的型攻撃 [1] が増加している．標的型攻撃では, 攻撃手法や用いられるマルウェアが特定組織に特化されている場合が多い．内向きの通信を監視する入口対策では攻撃を検知および防御することが困難になりつつある．その結果, 端末がマルウェアに感染する事を想定し, 内向きだけではなく外向きの通信ログから，攻撃の痕跡やマルウェアの活動を早期発見するための出口対策の重要性が増している．

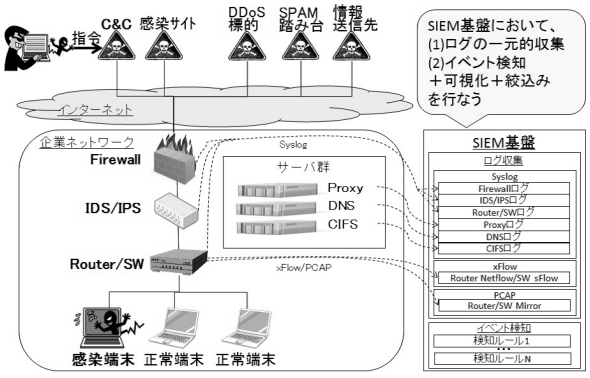

図 1: SIEM を用いた出口対策

出口対策の一つに図 1 に示すような SIEM を活用して収集した通信ログを分析し，マルウェアが行う不正な通信を検知するマルウェア感染端末検知技術がある．マルウェア感染端末検知技術においてはマルウェアの行う通信パターンを正確に定義しなければ誤検知が多くなってしまい，マルウェア感染端末を特定するのが難しいという問題がある．また，検知に用いる通信パターンの数が膨大になると分析処理時間が長くなってしまう．

上記の問題を解決するために本研究ではマルウェアの動的解析結果から得た通信ログを活用し，マルウェアの通信パターンを抽出する手法を提案する．具体的には各マルウェアの通信ログを複数の観点で一次分析した結果の時系列をマルウェア特有の通信パターンとみなす．検知精度の向上と処理時間低減のために特に複数のマルウェアで共通的にみられる通信パターンを抽出した．提案手法では誤検知を低減するために抽出した通信パターンに対して実網から収集した通信ログを用いてトレーニングを行う．また，提案手法によって生成された通信パターンを用いた検知精度評価を行った．その結果，誤検知率を 0.5%以下に抑えつつ，既存手法よりも 30%程度高い検知率を達成できることを確認した．

2　マルウェア感染端末検知技術における課題

2.1　既存の感染端末検知技術

出口対策では防御対象のネットワークの内側から外側への通信を監視し，C&C サーバとの通信や情報漏えいにつながるような不正な通信を検知する．近年では単一の機器だけではなく，様々な機器のログを SIEM によって収集し，収集したログを複数組み合わせて分析することで不正な通信の挙動を捉えるマルウェア感染端末検知技術がある．

感染端末検知技術の一つに外部への通信ログ, 具体的には Firewall と WebProxy から取得できる通信ログを活用して感染端末を検知する手法がある．通信ログの分析ではマルウェア感染

端末の行う不正な通信をルールによって検知する．検知精度を高めるためにマルウェアを動的解析した際に取得した通信ログから, 悪性サイトの URL の候補を抽出したり, 通信の振る舞いをルールとして生成する手法 [2] がある．

また，IDS のアラート情報を分析対象として利用して感染端末を検知する手法 [3] も Gu らによって研究されている．Gu らの手法では IDS のアラートのうち，特にマルウェアが行う通信が発生した際に現れるアラートを用いて，マルウェアの挙動とその遷移を感染モデルとして定義する．そして感染モデル内での状態の遷移の仕方をルールとして，監視対象ネットワークの通信を分析して感染端末の検知を行う．

既存手法における課題は以下の通りである．

(1) 検知精度：SIEM を活用して感染端末を検知する場合，少ない誤検知で，高い検知率を維持することが最も重要である．既存手法のようにマルウェアの通信ログから特徴を抽出し，ルールを生成して検知に用いる場合，検知率は非常に高くなる．しかし，マルウェアは解析を困難にするために正規のプログラムと同様の通信を行うことがある．そうした通信の影響を排除しなければ誤検知を引き起こす原因となる．誤検知を低減するためにはルールに対してチューニングが必須であるが，単一のルールの閾値をチューニングした結果，見逃しが増加し，全体的な検知率が低下してしまうという問題がある．

(2) 処理性能：検知精度に関しては個々のマルウェアに特化して検知可能なルールを用意することで，高い検知率を維持しつつ誤検知を低減させることが可能である．その一方で監視対象ログの分析時の処理時間がルール数に比例して増大するという問題がある．そのため，可能な限り少数の汎用的なルールで高い検知精度を得ることが課題である．

(3) チューニングの容易性：ルールを用いた通信ログ分析手法では誤検知を低減するために，監視対象ネットワークの環境にあわせて各ルールのチューニングが必須である．しかし，適切なチューニングを行うにはルールに関する知識が必要であり，各ルールの最適な閾値を得るために，複数回分析を実施しなければならず，チューニング作業に膨大な時間がかかる．また，日々進化し続けるマルウェアに対応するためには，ルールの追加や，定期的なチューニングも必須となり，さらに時間がかかってしまうという課題がある．

3 提案手法

本研究では前述の課題を解決するために，マルウェアの行う通信に共通的にみられる通信パターンの抽出手法について提案する．

3.1 マルウェア通信パターン抽出手法の概要

提案手法の概要を図 2 に示す．提案手法では，マルウェアの動的解析結果から得た通信ログをみなし悪性通信ログとする．マルウェア検体ごとに抽出したみなし悪性通信ログに対して，いくつかの通信の特徴を捉えたルールを用いて一次分析を行う．次に一次分析の結果からイベントの発生順序を基にイベント系列を生成する．このイベント系列を各マルウェアの行う通信パターンとみなす．時系列を考慮した通信パターンを検知に用いることで，単一のルールよりも正確にマルウェアの通信の特徴を捉えることができる．

生成した全ての通信パターンを分析に用いると分析処理時間が膨大になる．そのためイベント系列のクラスタリングを行い，類似したイベント系列をまとめ，各クラスタ内で共通的にみられるイベント系列を共通イベント系列として抽出する．抽出した共通イベント系列を検知に用いる検知用通信パターン候補とする．共通イベント系列の抽出によって検知に用いるイベント系列を削減することで分析処理時間を短縮できる．

2章でも述べたようにマルウェアは正規のプログラムと同様の通信を行うことがある．そのため，マルウェアの通信に含まれる正規の通信の影響を排除しなければ前述の手法で抽出された検知用通信パターンであっても誤検知が発生しうる．そこで提案手法では，実網には感染端末が少ないと仮定し，実網から収集した通信ログをみなし良性通信ログとして検知用通信パターン候補に対してトレーニングを行う．具体的にはみなし悪性通信ログと同様の一次分析とイベント系列の生成処理を行って，みなし良性通信ログイベント系列を生成し，検知用通信パターン候補とマッチングさせて一定割合以上でマッチしたものは検知用通信パターン候補から除外して検知用通信パターンを生成する．そのため，既存手法で行うような各ルールのチューニングが必要がない．

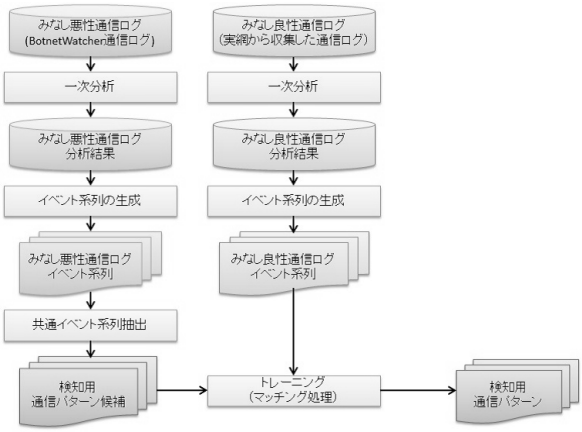

図 2: マルウェア通信パターン抽出手法の概要

3.2 ルールによる一次分析

提案手法では各マルウェアの通信ログをルールによって一次分析した結果をイベントと呼び，イベントの出現の仕方を各マルウェアの通信パターンと定義する．

図3に提案手法で利用するルールの生成手法について図示する．ハニーポットやマルウェア共有サイトから収集したマルウェアの動的解析を行い，マルウェアを一定期間動作させることで蓄積した通信ログを解析してルールの生成を行う．動的解析には BotnetWatcher[4] を用いた．通信ログにはマルウェアが外部との通信を行った際の Firewall と WebProxy のログが含まれる．Firewall のログからは 5-tuple を中心としたレイヤ 4 までの情報を基に分析する IP 系ルールを生成する．WebProxy のログからは HTTP 通信に含まれる URL や UserAgent の情報を基に分析する HTTP 系ルールを生成する．

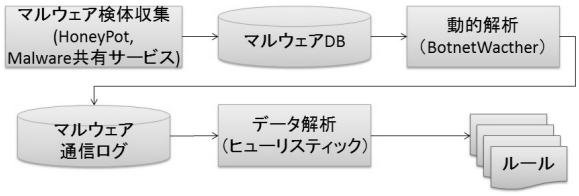

図 3: ルール生成手法

ルール生成においては単独のマルウェアではなく，複数のマルウェアで見られる特徴を次の4つルールのタイプに分けて定義し，全 31 ルールを利用して一次分析を行った．また，各レイヤのタイプごとのルール数について表1に示す．

- 悪性リスト型
  マルウェアの通信に頻繁に見られる URL や IP アドレス，ポート番号，UserAgent を含む通信の回数をパラメータとして数え上げるルール．
- 異常通信型
  正規の通信ではあまり見られない通信の回数をパラメータとして数え上げるルール．
- スキャン型
  異常通信型の中でも短期間で宛先アドレス/ポートの種類数をパラメータとして数え上げるルール．
- 発生間隔型
  送受信 IP アドレスが同一条件の通信で発生間隔に周期性が見られる通信の回数をパラメータとして数え上げるルール．

一次分析の結果には，srcIP アドレス（送信元 IP アドレス）単位で，一定期間内に検知されたルール名とそのルールで数え上げられた回

表 1: ルール分類一覧

対象レイヤ	ルールタイプ	ルール数
IP	悪性リスト型	7
	異常通信型	4
	スキャン型	6
	発生間隔型	2
HTTP	悪性リスト型	7
	異常通信型	4

数および，検知時刻が記録される．ルールごとに数え上げられた数をそのルールのスコアとする．なお，マルウェア検体を動的解析する際に一意に srcIP アドレスを付与して分析した．

3.3 イベント系列の生成

3.2 節で述べたルールを用いた一次分析の結果を基に各マルウェアのイベント系列を生成する．イベント系列生成手法の概要を図 4 に示す．まず，srcIP アドレス単位で一次分析結果を収集し，ルール名とスコアのセットを一つのイベントとする．イベント系列の候補内で 2 つのイベントの検知時刻が所定のイベント系列分割間隔以上離れている場合は，後ろのイベントから残りの部分を新たに別のイベント系列候補として分割する．提案手法では通信の特徴をあらわす各イベントの発生の仕方に着目してイベント系列を生成する．同一イベントの発生回数は環境に依存して変わりうるため，検知時に回数の違いによって見逃しが発生する可能性がある．そのため，1 つのイベント系列候補内に同一イベントが存在する場合は，2 回目以降の同一イベントを除去した上でイベント系列とする．

3.4 共通イベント系列抽出

個々のマルウェアから生成された全てのイベント系列を検知に用いると，マルウェア検体の数に比例して分析時の処理時間が増大してしまう．提案手法では生成されたイベント系列に対してクラスタリングを行い，類似のイベント系列を

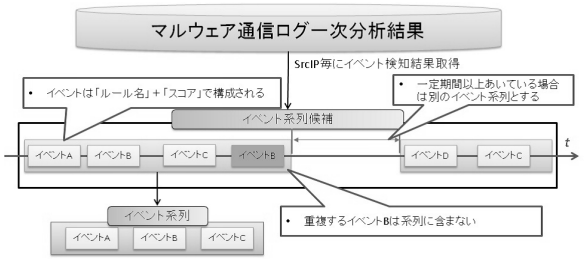

図 4: イベント系列生成手法の概要

まとめ，クラスタ内で共通的にみられるイベント系列を抽出する．共通的なイベント系列にまとめることで，分析時の処理時間を抑えると同時に，振る舞いが一部異なる亜種についても幅広く検知することが可能となる．提案手法ではクラスタリングには fastcluster[5] を用いた．クラスタリングに用いる距離関数 d は Levenshtein 距離を用いて以下のように計算する．

$$A, B : \text{イベント系列}$$
$$Len(A) : A \text{ に含まれるイベント数}$$
$$LD\,(A, B) : A, B \text{ 間の } Levenshtein \text{ 距離}$$
$$d\,(A, B) = \frac{LD(A, B)}{max\,(Len(A), Len(B))} \quad (1)$$

次に共通イベント系列の抽出手法について述べる．共通イベント系列は，クラスタに所属するイベント系列が少なくとも 3 以上であるクラスタから抽出する．クラスタに所属するユニークなイベント系列が 1 つの場合は，そのイベント系列を共通イベント系列として抽出する．クラスタに所属するユニークなイベント系列が 2 つ以上存在する場合には，クラスタ内のイベント系列を 2 つ取り出し，全てのペアにおいて LCS（Longest Common Subsequence）を求める．抽出した共通イベント系列のうち，短すぎるものは通信パターンとは言えないため，最小共通イベント系列長以上のものを検知用通信パターン候補とする．

3.5 トレーニング

検知用通信パターン候補に含まれる共通イベント系列から，正規の通信による影響を除外す

るためトレーニングを行う．まず，マルウェアの通信ログと同様に実網の通信ログからみなし良性通信ログイベント系列を生成する．次に生成したみなし良性通信ログイベント系列と検知用通信パターン候補とマッチングを行う．マッチング処理ではみなし良性通信ログイベント系列に含まれるイベント系列 ES_A と，検知用通信パターン候補に含まれる共通イベント系列 CS_B と間で LCS を求める．$LCS\,(ES_A, CS_B)$ と共通イベント系列 B の長さ $Len(CS_B)$ から以下のようにマッチ率を求める．

$$match\_ratio = \frac{Len(LCS(A,B))}{Len(CS_B)} \tag{2}$$

マッチ率が所定のマッチ率閾値以上になった共通イベント系列を検知用通信パターン候補から除外する．トレーニングを行った結果，残った共通イベント系列を検知用通信パターンとする．

4　提案手法の評価実験

提案手法によって抽出した通信パターンを用いて，マルウェアの通信ログと，実網から収集した通信ログの分析を行うことで提案手法の有効性を検証した．

4.1　評価指標

4.1.1　検知精度

検知精度を確認するために，提案手法で生成された検知用通信パターンを用いた分析を行い，検知率・誤検知率を測定した．比較対象として一次分析のルールを単独で適用して分析する場合を既存手法とし，同様の評価用データを用いて検知率・誤検知率を測定した．検知率と誤検知率は以下の式で算出する．

TP = 正しく検知されたユニーク $srcIP$ 数

FN = 誤って検知されたユニーク $srcIP$ 数

$$\text{検知率} = \frac{TP}{srcIPs_m} \tag{3}$$

$$\text{誤検知率} = \frac{FN}{srcIPs_l} \tag{4}$$

表 2: 評価用データ

ログ種別	レコード数	SrcIP 数
みなし悪性通信ログ	3,167,315	25,049
みなし良性通信ログ（Firewall）	12,064,822	8,570
みなし良性通信ログ（WebProxy）	20,911,294	2,060

$srcIPs_{\mathrm{m}}$ はみなし悪性通信ログに含まれる全 srcIP アドレス数を，$srcIPs_l$ はみなし良性通信ログに含まれる全 srcIP アドレス数を示す．なお，既存手法の各ルールは，ルール単体の誤検知率が 0.5%以下になるようにチューニング [6] を行った．

4.1.2　処理性能

提案手法によって生成された通信パターンを用いて感染端末検知を行う場合の処理性能について測定を行った．処理性能は生成処理と分析処理について測定した．生成処理時間は検知用通信パターンを生成が完了するまでの時間である．分析処理時間は生成した検知用通信パターンを用いて評価用データの分析が完了するまでの時間である．なお，処理性能に関しては以下の理由により既存手法との比較は行わない．生成処理にかかる時間は既存手法におけるチューニングにかかる時間が比較対象となる．しかし，既存手法のチューニングは誤検知の発生状況により変動が激しい．そのため，比較対象となる処理時間を一意に決めることができない．また，分析処理の時間に関しては，提案手法は既存手法と同様の手順で一次分析を行っている．前述の分析処理時間は，既存手法でかかる分析処理時間に単純に追加される時間であり比較できない．

4.2　評価用データ

評価実験には表 2 に示したみなし悪性通信ログとみなし良性通信ログを用いた．みなし悪性

通信ログは VirusTotal[7] から 35 日分のマルウェア検体を取得し，BotnetWatcher を用いて動的解析結果から得た通信ログであり，検知用通信パターンの生成と検知率の評価に用いた．みなし良性通信ログは実網から収集した Firewall 及び WebProxy の 7 日分の通信ログであり，検知用通信パターン候補のトレーニングおよび誤検知率の評価に用いた．なお，以降の実験では評価データを 3 分割してクロスバリデーションを実施した．

4.3 実験条件

4.3.1 パラメータ設定

提案手法における各種パラメータを表 3 の通りに設定し，以降の評価実験を通じて変更しなかった．

表 3: パラメータ設定値

パラメータ名	値
イベント系列分割間隔（秒）	3600
最小共通イベント系列長	4
マッチ率閾値（%）	80

4.3.2 実験環境

検知用通信パターンの生成および評価データを用いた分析は，CPU が Intel(R) Xeon(R) CPU E5-2403@1.80GHz の 4 コア，メモリ容量 48GB の IA サーバで実施した．

4.3.3 検証パターン

提案手法に関して，検知精度及び性能処理への影響を測定するために，3.4 節で定義した距離関数 d の値 3 種と，3.5 節で説明したトレーニングの有無によって表 4 に示した 6 パターンで実験を行った．

表 4: 検証パターン

	d の値	トレーニング
Case1	0.1	有
Case2	0.1	無
Case3	0.2	有
Case4	0.2	無
Case5	0.3	有
Case6	0.3	無

5 評価結果

5.1 検知精度評価結果

各検証パターンにおける検知率，誤検知率の評価結果を表 5 に示す．既存手法における誤検知率は各ルールで誤検知された srcIP を集計し，ユニーク数にして計算している．

表 5: 検知精度評価結果

	検知率	誤検知率
既存手法	44.90 %	3.20 %
Case1	71.10 %	0.17 %
Case2	72.80 %	4.92 %
Case3	76.23 %	0.20 %
Case4	78.09 %	8.80 %
Case5	79.47 %	0.27 %
Case6	81.03 %	9.76 %

5.2 処理性能評価結果

各検証パターンにおける処理性能の評価結果を表 6 に示す．生成時間および分析時間に加えて，処理性能への影響を考察するために，生成された検知用通信パターン数もあわせて記載する．

6 考察

検知率に関しては提案手法はどのケースにおいても既存手法を 30%前後上回る結果となった．

表 6: 処理性能評価結果

	パターン数	生成時間	分析時間
Case1	727	0:35:08	0:19:16
Case2	766	0:16:02	0:13:55
Case3	1,332	0:57:32	1:00:01
Case4	1,386	0:12:31	0:26:45
Case5	2,415	1:51:24	2:20:46
Case6	2,512	0:16:16	1:00:09

誤検知率はトレーニングを実施した Case1,3,5 では検知率が 0.5%以下となった．既存手法では各ルールを誤検知率 0.5%でチューニングを実施しているため，提案手法は全体の誤検知率・単一ルールの誤検知率のどちらと比べても誤検知率が低いと言える．また，トレーニングの有無による誤検知率の変動の大きさに対して，検知率の変動は 2%前後であり，提案手法におけるトレーニングの有効性が証明できた．

次に処理性能に関して考察を述べる．クラスタリングに用いる d の値を大きくすることで検知用通信パターン数が増加する，また，トレーニングを行わない場合には生成処理時間は短縮される．一方で除外されるべき共通イベント系列が除外されないため，検知用通信パターン数が増加する．理論上は検知用通信パターン数が増加することで，分析時にマッチング対象が増えるため分析処理が長くなると想定されるが，今回の実験では短くなっている．今回の実験で用いたプログラムではイベント系列の作成からトレーニング後の分析まで通して実行している．そのためトレーニングの有無によって LCS の計算結果のキャッシュ内容の差が影響している可能性がある．しかし，入力データ量が 7 日分のデータを 3 分割したものであること考慮すると，最長時間の Case5 であっても日次で分析を行う場合には運用可能な処理時間だと言える．

7 おわりに

本研究ではマルウェアの動的解析結果からマルウェアの通信に共通的にみられる通信パターンの抽出手法を提案した．提案手法の評価をマルウェア通信ログと実網から収集した通信ログを用いて実施した．評価結果から既存手法よりも誤検知を低減しつつ高い検知率を維持できることを確認した．また，分析処理性能についても日次分析に適用可能であることを確認した．今後は実験用プログラムを改良した上で関連研究である BotHunter 等との検知精度及び処理性能の比較を行う予定である．

参考文献

[1] 情報処理推進機構, “「新しいタイプの攻撃」の対策に向けた設計・運用ガイド 改訂第 2 版,” http://www.ipa.go.jp/files/000017308.pdf,Nov.2011.

[2] 中田健介，“ネットワークログ分析による不正通信検出手法” 電子情報通信学会総合大会, 通信講演論文集 No.2,pp.163，Mar. 2013.

[3] Guofei Gu et al. , “ BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation ” USENIX security 2007.

[4] K. Aoki, T. Yagi, M. Iwamura, and M. Itoh, “ Controlling Malware HTTP Communications in Dynamic Analysis System Using Search Engine, ” 2011 Third International Workshop on Cyberspace Safety and Security (CSS), pp.1—6, Sep. 2011.

[5] Daniel Müllner, fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python, Journal of Statistical Software 53 (2013), no. 9, 1-18, URL http://www.jstatsoft.org/v53/i09/

[6] 中田健介，佐藤徹，倉上弘，角田進，“ネットワークログ分析における不正通信検知ルールの最適化手法” 電子情報通信学会総合大会, 通信講演論文集 No.2,pp.243，Mar．2015.

[7] VirusTotal，https://www.virustotal.com/ja/