マルウェアの通信特徴に基づく SSL/TLS を利用した C&C 通信検出手法

寺田 成吾† 小林 峻† 道根 慶治† 山下 康一†

概要：マルウェアを利用したサイバー攻撃では，Command and Control サーバーとのネットワーク通信（C&C 通信）を介して遠隔でマルウェアを操作する手法が用いられる．そのため，マルウェアに感染した端末を検出する有効な手段の 1 つとして，C&C 通信の特徴をもとに検出する手法がある．しかし，C&C 通信に SSL/TLS を利用し，C&C 通信のシグネチャ等の特徴を隠ぺいすることでセキュリティ機器による検出を回避する攻撃がある．さらに，SSL サーバー証明書として自己署名証明書が攻撃に利用されていたが，昨今の HTTPS 通信の普及に伴い，無料で正規の SSL サーバー証明書が利用できるようになったため，SSL/TLS を利用する攻撃が増加している．本研究では，組織内の端末が送受信する SSL/TLS 通信を監視し，送信間隔の規則性の度合いと受信データサイズの偏りの度合いといったマルウェアの通信にみられる特徴から SSL/TLS を利用した C&C 通信を復号化することなく検出する手法を提案する．

キーワード：ネットワーク，C&C 通信，SSL/TLS，送信間隔，受信データサイズ

the Detection Method for C&C Communication using SSL/TLS based on Characteristics of Malware Communication

Seigo Terada† Takashi Kobayashi† Keiji Michine† Koichi Yamashita†

Abstract: In cyber threats using malware, an attacker remote control a malware through network communication with Command and Control server (C&C communication). Thus the detection method based on the characteristics of C&C communication for infected machine is one of the effective means. But there are attacks hide characteristics such as signature of C&C communication by using SSL/TLS to avoid a detection by security system. Furthermore, in previous attack, some attackers use self-signed certificates as SSL server certificate, recently, some attackers use free authorized certificates due to the spread of adoption of HTTPS. In this paper, we propose the detection method for C&C communication using SSL/TLS without decrypting by watching SSL/TLS communication by computer machine and analyzing regularity of request interval and bias of response data size.

Keywords: network, C&C communication, SSL/TLS, request interval, response data size

1. はじめに

標的型攻撃等によるインシデントが継続して発生している．標的型攻撃のような高度な攻撃では，標的型メールや水飲み場攻撃といった手法で組織ネットワーク内にマルウェアを侵入させる．初期の侵入段階では，文書ファイルのマクロ機能を悪用したマクロウィルスやオンラインストレージからマルウェアをダウンロードさせるなど様々な手法を攻撃者は利用する [1]が，最終的に組織内のコンピューター端末に遠隔操作可能な RAT（Remote Access Trojan）を感染させて長期的に組織ネットワークを侵害する攻撃基盤を形成する．標的型攻撃において，セキュリティ侵害が発生してから検出に要する日数の中央値は，約 100 日という調査結果 [2]があり，一度侵入されるとセキュリティ侵害が長期化するケースがある．

初期の侵入段階での攻撃経路は多く，また，修正情報が公開されていないゼロデイ脆弱性を利用することもあるため，近年のサイバーセキュリティ対策では侵入されることを前提とした内部対策の重要性が主張されている [3] [4]．侵入された後の内部対策では，マルウェアから外部の攻撃者のサーバーである Command and Control サーバー（C&C サーバー）との通信（C&C 通信）を検出するために組織内の通信を監視することが重要である．なぜなら，C&C 通信はマルウェアを利用した攻撃基盤の重要な要素となっており，遠隔からマルウェアに指令を与えることで機密情報の窃取，マルウェアのバイナリのアップデートや設定ファイルの変更等，柔軟に攻撃活動を行なえるように設計されているため，C&C 通信を検出できれば，組織内部に潜むマルウェアを検出できるからである．

マルウェアの C&C 通信では，ファイアウォールを通過することができる 80 番ポート，443 番ポートを 8 割のマルウェアが利用し，また，通信プロトコルとして HTTP または HTTPS（HTTP over SSL/TLS）を利用する．トレンドマイクロ株式会社の調査では，その割合は 2014 年から 2017 年で傾向に変化はないように見られる [5] [6]．これらの通信プロトコルを利用する理由としては，組織の業務通信の中に

† 株式会社 PFU
PFU Limited.

C&C 通信を紛れ込ませ，検出を困難にさせる目的があると思われる．特に HTTPS のように SSL/TLS を利用した C&C 通信は，特徴的なシグネチャを暗号通信の中に隠ぺいできるため，平文で通信される HTTP より検出が困難になる．SSL Blacklist [7]で C&C 通信に利用される SSL サーバー証明書を確認してみると，多くは SSL サーバー証明書として自己署名証明書が利用されているが，昨今の HTTPS 通信の普及に伴い，無料で正規の SSL サーバー証明書が利用できるようになったため，PandaZeus など C&C サーバーに正規のサーバー証明書を利用するマルウェアの増加が伺える．SSL/TLS を利用した通信を検査する方法として，SSL インスペクションのように組織出口の通信経路上で暗号通信を一度復号し，通信の中身を検査する方法がある．しかし，機器の導入コスト，運用コスト，通信性能（復号化/暗号化処理コスト），プライバシーを保護できない問題がある．

本研究では，SSL/TLS 通信を復号化することなくリアルタイムに C&C 通信検出する新たな手法を提案する．本手法では，まず，端末が送信する ClientHello を SSL/TLS Fingerprinting [8] [9]により主要ブラウザの ClientHello との類似性を評価し，主要なブラウザに類似しない通信を検査対象として抽出する．主要ブラウザによる通信を除外したのち，検査対象となった同じ宛先に対して送信する ClientHello の送信間隔の規則性，暗号通信のサーバーからクライアント方向の Application Data のサイズの偏り度合いを算出し，C&C 通信か否か判定する．本手法では，マルウェアの C&C 通信は，攻撃者からの指令を待ち受ける段階において，ClientHello を送信する期間には規則性があり，レスポンスデータサイズは一定の偏りが見られると仮定している．この仮定は，端末への侵入直後や攻撃者からの指令を待ち受けているようなマルウェアが攻撃者からの指令がない状態では，新たな指令がないか定期的に C&C サーバーの情報を確認するような通信を行い，指令に変化がない場合，C&C 通信のレスポンスデータは，一定の値を返すという一連の動きを想定している．

本手法を評価するためにマルウェアが実際に動作した際の通信データが公開されているデータセット，およびインターネット上でサービス提供しているサンドボックス環境での動作結果を収集し，一定数以上の ClientHello を送信しているキャプチャデータを選別した．

本論文の構成は，2 章では関連研究を紹介し，3 章で本稿における C&C 通信検出手法について述べた後に，4 章で本手法の評価結果を記述する．最後に 5 章でまとめと今後の課題について述べる．

2. 関連研究

B. Anderson ら [10]は，TLS 通信，DNS 通信，HTTP ヘッダーと SSL/TLS の暗号化前の情報でマルウェアによる通信と正常な通信を識別する手法を提案している．また，その後の研究 [11]では，SSL/TLS ハンドシェイクに見られる特徴に着目して C&C 通信における SSL/TLS の利用形態を明らかにし，マルウェア・ファミリーの分類を試みている．どちらの研究においても SSL/TLS ハンドシェイクにおける ClientHello や ServerHello，サーバー証明書といった情報を Fingerprint として C&C 通信の特徴を検出している．

Frantisek Strasak [12]は，Bro IDS [13]で得られるログファイルから 4-tuple 情報（送信元 IP，送信先 IP，送信先ポート，プロトコル）と SSL/TLS ハンドシェイク，サーバー証明書から得られる情報から 28 個の特徴量を定義し，複数の機械学習アルゴリズムを利用することで SSL/TLS を利用した C&C 通信を復号することなく検出する手法を提案している．

本手法では SSL/TLS Fingerprinting は正規ブラウザによる通信を除外して誤検出を押さえるために利用している．そのため，未知の SSL/TLS Fingerprinting を利用した C&C 通信でも検出可能である．また，SSL/TLS Fingerprinting に用いる情報は，ClientHello から得られる情報に留めており，ServerHello やサーバー証明書の情報は利用していない．特にサーバー証明書は，SSL/TLS のセッションの再利用時にダウンロードされない場合があるため，C&C 通信の特徴として採り上げていない．本手法は，ClientHello の送信間隔の規則性，レスポンスデータサイズの偏りのみに着目することで SSL/TLS を利用した C&C 通信を復号することなく検出可能な手法を目指している．

3. 提案手法

3.1 SSL/TLS Fingerprinting によるフィルタリング

SSL/TLS Fingerprinting は，SSL/TLS 通信において暗号通信を行なう前段階の SSL/TLS ハンドシェイクにおいてクライアントが送信する ClientHello に含まれる Cipher Suite リストや TLS extensions リストなど観測可能な情報を元にクライアントアプリケーションを推定する技術である．主要ブラウザとマルウェアの SSL/TLS 通信の異なる点として，利用する ClientHello Extensions がある．ブラウザが汎用的で高機能ゆえに様々な ClientHello Extensions を利用するのに対し，マルウェアは特定用途であり，汎用あるいは自作の SSL/TLS ライブラリを利用するため，限られた ClientHello Extensions を利用する．

本手法では，SSL/TLS Fingerprinting を判別する要素として，ClientHello Extensions の数と種類，ECDHE/FFDHE グループのリストの要素数，Ciphersuites に含まれる暗号化アルゴリズムの要素数を定義し，各値がそれぞれ一致するか否かによって主要ブラウザとの通信かを判定する．

3.2 ClientHello 送信間隔の規則性

本手法では，ClientHello の送信間隔の規則性を判定するために度数分布のクラス分割（区間分割）の考え方を利用

した．ClientHello の送信間隔を複数のクラスに分割し，送信間隔の分布散布度と個々のクラスに属する送信間隔のばらつき度合いから，ClientHello の送信間隔の規則性を分析する．この規則性の度合いを RoI（Regularity of Interval）と定義し，RoI は値が大きいほど規則性の度合いが高いことを表す．ClientHello の送信間隔は，図 1 に示すとおり ClientHello の送信時刻の差分とし，分析を行なう閾値としてσ 回分の送信間隔を集めるため，σ + 1回の ClientHello を記録する．

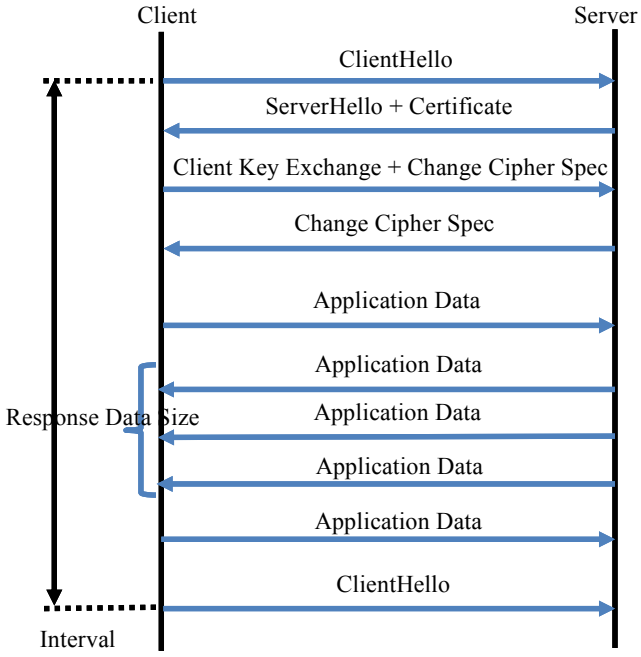

図 1 ClientHello の送信間隔とレスポンスデータサイズ

以下に RoI を求める手順を示す．

(1) ClientHello X_i と X_{i+1}（i=1～σ）の送信間隔 I_i の集合 $I_{interval}$ を作成する．

(2) k-means を用いて $I_{interval}$ を N_C 個のクラスに分割する．クラスの分割数 N_C は，Sturges の公式 [14]を用いて以下のとおり設定する．

$$N_C = 1 + \log_2 \sigma$$

また，k-means の初期値は，結果がランダムになるのを避けるため，$I_{interval}$ を昇順にソートし，固定の要素サイズの重心を N_C 個計算して初期値とした．クラス分割したのち，各クラスの送信間隔の平均値が近しいクラスは 1 つのクラスとしてマージした．

以降の分析は，クラスの要素数が 4 個を越える

(3) クラス C_n 内で送信間隔の平均値を $Avg(C_n)$ より計算し，誤差の許容上限値 $S_{up}(C_n)$ と許容下限値 $I_{nf}(C_n)$ を定める．ここで，$Avg(C_n)$ は，集合の最大と最小の要素を除外した平均値を求める関数であり，$\theta(Avg)$ は，平均値に対応した許容誤差を返す関数である．

$$Avg(C_n) = \frac{\sum C_n - \left(\max(C_n) + \min(C_n)\right)}{|C_n| - 2}$$

$$S_{up}(C_n) = \left\{I_i \in C_n | \, Avg(C_n) * \left(1 + \theta\left(Avg(C_n)\right)\right)\right\}$$

$$I_{nf}(C_n) = \left\{I_i \in C_n \mid Avg(C_n) * \left(1 - \theta\left(Avg(C_n)\right)\right)\right\}$$

(4) クラス C_n ごとに許容の範囲内に存在する送信間隔の集合 $\mathrm{T_{ol}}$ とその要素数 $|\mathrm{T_{ol}}|$ を求める．

$$\mathrm{T_{ol}}(C_n) = \{I_i \in C_n | \, I_{nf}(C_n) \le I_i \le S_{up}(C_n), |C_n| > 0\}$$

(5) 送信間隔の度合い RoI を以下の式で求める．

$$\mathrm{RoI} = \min\left\{\mathrm{RoI}(C_n) | \mathrm{RoI}(C_n) = \frac{|T_{ol}(C_n)|}{|C_n|}, \forall |C_n| \ge 4\right\},$$

$$0 \le \mathrm{RoI} \le 1.0$$

3.3 レスポンスデータサイズの偏り

レスポンスデータサイズは，クライアント方向への一連の Application Data の合計サイズとしている．このレスポンスデータサイズの偏りの度合いを BoR（Bias of Response data length）と定義し，ある宛先への ClientHello に対するレスポンスデータサイズの偏り度合いを評価する．BoR は，値が大きいほど偏りの度合いが高いことを表す．本手法では分析対象を C&C 通信としているため，マルウェア自身のアップデートや設定のアップデートのためにバイナリをダウンロードする通信はノイズとなりうるため，分析対象から除外したい．そのため，レスポンスデータサイズが 64Kbyte を超えていた場合，分析対象から除外することとした．以下に BoR を求める手順を示す．

(1) ClientHello X_i に対するレスポンスのデータ長 $RL(X_I)$ を δ 個集め，集合 RL とする．

(2) レスポンスデータ長の集合 RL の各要素に対する上限閾値 SRL を以下の式をもとに定める．

$$SRL = 1.2 * \left(\frac{\sum_i RL(X_i)}{\delta}\right), RL(X_i) \in RL$$

(3) ClientHello X_i に対するレスポンスのデータ長が SRL 以下である X_i からなる集合 TR_{SRL} を求める．

$$TR_{SRL} = \{RL(X_i) \le SRL, \; RL(X_i) \in RL\}$$

(4) TR_{SRL} の要素数 $|TR_{SRL}|$ をもとに以下の式より BoR を求める．

$$BoR = \frac{|TR_{SRL}|}{\delta},$$

$$0 \le BoR \le 1.0$$

4. 評価

本手法の評価として，C&C 通信の検出率評価，通常通信を利用した誤検出評価を行った．本試験で用いたパラメータは, 集める送信間隔数の閾値をσ, C&C 通信と見做す RoI, BoR の閾値をそれぞれρ，μとして，$\sigma = 16$，$\rho = 0.6$，$\mu = 0.7$を用いた. $\sigma = 16$より, 送信間隔の分割数N_Cは, $N_C = 1 + \log_2 16 = 5$となる．

4.1 評価環境

ネットワーク通信をパッシブで監視し，リアルタイムで解析するプログラムとして実装しているが，効率的に評価するために解析プログラムにキャプチャファイルを直接読み込む改造を行い，タイムスタンプを基準にパケットを処理するプログラムを作成した．このプログラムに評価対象のキャプチャファイルを順番に処理させた．

4.2 検出性能の評価

4.2.1 公開データセット

公開されているデータセットとして CTU データセット [15]，Malware-Traffic-Analysis.net [16]，また，オープンな解析サービスである Hybrid Analysis [17]からパケットキャプチャファイル（pcap）をダウンロードし，ClientHello 数が規定数あるファイルを選別して今回用いるデータセットとした．各データセットの pcap ファイル数を表 1 に示す．各データセットに含まれるマルウェア・ファミリーとそのデータ数は，付録に記載する．

表 1 公開データセット一覧

データセット	pcap ファイル数
CTU Dataset (Malware)	27
Malware-Traffic-Analysis.net	51
Hybrid Analysis	22
合計	100

4.2.2 検出性能の評価結果

表 2 に評価結果を示す. 検出の可否は, 閾値σの数を集めた ClientHello の集合のうち，少なくとも 1 つを C&C 通信とみなしたか否かで判定した．

表 2 検出率評価の結果

データセット	pcap ファイル数	検出数
CTU Dataset (Malware)	27	27
Malware-Traffic-Analysis.net	51	48
Hybrid Analysis	22	19
合計	100	94 (94.0%)

一部検出できていない C&C 通信があり, 検出できていないパターンとしては，以下の 2 パターンがあった．

1. 平均的に 500～600 バイトのレスポンスデータを受信しているが，ときおり 0 バイトほどの小さなレスポンスデータを複数回受信し，分析時に平均値を算出すると 300～400 バイトになり，*SRL* が頻出する 500～600 バイトより小さくなるため, BoR が小さくなるパターン．
2. 送信間隔のクラス分割がうまくいかず，あるクラスにおいて，値がかけ離れた複数の要素を 1 つのクラスとして扱ってしまうために RoI の値が 0.0 付近になるパターン．

4.3 正常通信による誤検出評価

CTU-Dataset の Normal データセット [18]を用いて正常通信を C&C 通信と誤検出する性能を評価した．Normal データセットは，Alexa top 1000 内で HTTPS に対応しているサイトにアクセスした際の通信を記録した pcap ファイルである．アクセスするブラウザとして，Firefox®[a]および Iceweasel が利用されている．表 3 に利用したデータセットの情報を記載する．各データの対象宛先 IP 数は，全宛先 IP 数の中で ClientHello の送信回数が閾値σ を越えている送信先の数を示し，ClientHello 数は，データ内に記録されている ClientHello の数である．

表 3 CTU-Normal データセット

データ番号	対象宛先 IP 数（全宛先 IP 数）	ClientHello 数
CTU-Normal-29	40 (1130)	5275
CTU-Normal-30	92 (1539)	8545
CTU-Normal-31	101 (1786)	9674
CTU-Normal-32	107 (1840)	10,248
合計	340 (6295)	33,742

まず，SSL/TLS Fingerprinting によるブラウザ通信の除外を行なった評価結果を表 4 に示す．Firefox®，Iceweasel に対して，ともに同じ SSL/TLS Fingerprinting の設定で，ほぼすべての HTTPS 通信をブラウザによる SSL/TLS 通信と判定し，分析対象外としていた．パケット情報が一部壊れたと思われる SSL/TLS のトランザクションの解析おいて，ClientHello の SSL/TLS Fingerprinting の識別に失敗しており，ブラウザ以外のアプリケーションによる通信として分析対象になり，収集されていたが，収集した数が閾値σ を越えることが無かったため, C&C 通信分析処理は実行されなかった．

a) Firefox は，Mozilla Foundation の登録商標です．

表 4 誤検出評価結果（SSL/TLS Fingerprinting あり）

データ番号	対象宛先 IP 数（全宛先 IP 数）	検出	誤り率 (%)
CTU-Normal-29	40 (1130)	0	0.0
CTU-Normal-30	92 (1539)	0	0.0
CTU-Normal-31	101 (1786)	0	0.0
CTU-Normal-32	107 (1840)	0	0.0
合計	340 (6295)	0	0.0

次に，SSL/TLS Fingerprinting によるブラウザ通信の除外の影響を把握するために，ブラウザ通信を除外せずに行なった評価結果を表 5 に示す．平均で対象宛先 IP への通信のうち 33.2%を C&C 通信だと誤って判定していた．ClientHello 数が閾値σに満たなかった宛先 IP への通信を含めた全宛先 IP への通信で計算すると 1.80%を C&C 通信と誤って判定していた．よって，本手法では SSL/TLS Fingerprinting によるブラウザ通信の除外が実用上必要である．

表 5 誤検出評価結果（SSL/TLS Fingerprinting なし）

データ番号	対象宛先 IP 数（全宛先 IP 数）	検出	誤り率 (%)
CTU-Normal-29	40 (1130)	9	22.5 (7.96)
CTU-Normal-30	92 (1539)	40	43.5 (2.60)
CTU-Normal-31	101 (1786)	27	26.7 (1.51)
CTU-Normal-32	107 (1840)	37	34.6 (2.01)
合計	340 (6295)	113	33.2 (1.80)

4.4 評価結果の考察

図 2 に検出性能評価において，C&C 通信を正しく検出できた結果（TP），C&C 通信を検出できなかった結果（FN），誤検出評価において，SSL/TLS Fingerprinting を用いてブラウザ通信を分析除外しない場合に，正常通信を C&C 通信と検出した結果（FP），正常通信を C&C 通信として検出しなかった結果（TN），それぞれの RoI 値，BoR 値をプロットした散布図を示す．TP，TN の違いは，RoI に大きく表れており，特に RoI が 1.0 である場合は TP，RoI が 0.0 である場合は TN である割合が大きく表れている．BoR の値は，今回用いた閾値$\mu = 0.7$を上回る通信の割合が多く，今回の手法ではうまく偏りを分析できていないため，C&C 通信と正常通信を判別できるほど有意な特徴になっていない．

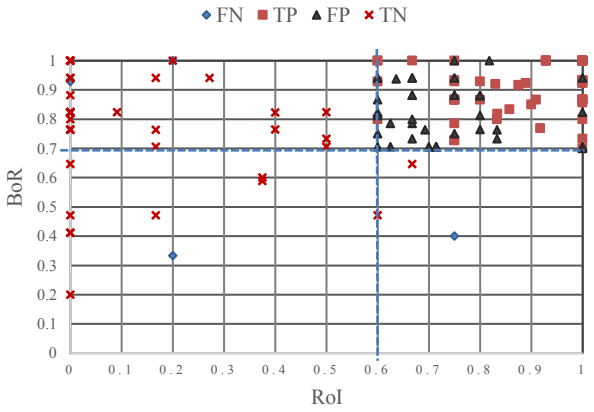

図 2 評価結果の散布図（RoI，BoR）

5. まとめと今後の課題

SSL/TLS を利用した C&C 通信検出手法を提案した．検出率評価において，100 個のデータに対して，94.0%の検出率であった．よって，本手法が提案する ClientHello の送信間隔の規則性と Application Data のレスポンスデータサイズの偏りに着目することは，マルウェアの C&C 通信の検出に有効である．一方，誤検出の評価では，ClientHello 数が閾値を超えた場合，平均で 33.2%の通信を C&C 通信と誤検出してしまう．この誤検出は，SSL/TLS Fingerprinting を用いて正規ブラウザ通信を判別し，分析除外とすることにより，誤検出を防ぐことができることを確認した．SSL/TLS Fingerprinting による除外処理は，今回評価したマルウェア・ファミリーの C&C 通信に対して，過度に C&C 通信を除外対象とする見逃しはなかったため，有効な誤検出対策であった．

今後の課題として，SSL/TLS を用いた C&C 通信を行なうマルウェアのデータセットが不足しているため，継続してデータ入手と評価を行ない，手法の有効性を確認していく．誤検出評価において，SSL/TLS Fingerprinting による除外がない場合，C&C 通信と正規通信の分離がうまくいっていなかった．BoR の値が C&C 通信と正規通信で大きな差異が無かったため，まず BoR の分析方法を見直すことで，改善を試みる．また，RoI について極端に短い期間に ClientHello を複数回送信するブラウザの通信を規則性ありとして判定する場合があったため，ブラウザのこの特徴を考慮した見直しを考える．最後に今回の誤検出評価は単一環境で限られたブラウザを用いたデータセットを用いたため，より複数のブラウザやアプリケーションが混在する実環境に近いネットワーク環境における誤検出評価が必要である．そして，誤検出評価で追加となる SSL/TLS Fingerprinting と C&C 検出性能への影響も把握していく必要がある．

参照文献

[1] 独立行政法人情報処理推進機構, “サイバー情報共有イニシアティブ（J-CSIP） 運用状況 [2018 年 1 月～3 月],” 2018.

[2] MANDIANT, M-TRENDS 2018, 2018.

[3] 株式会社ラック, “サイバー救急センターレポート - 脅威管理とインシデント対応をする人へ- 第 3 号,” 2018.

[4] 独立行政法人情報処理推進機構, 情報セキュリティ白書 2018, 2018.

[5] トレンドマイクロ株式会社, “国内標的型サイバー攻撃分析レポート 2015 年版～「気付けない攻撃」の高度化が進む～,” 2015.

[6] トレンドマイクロ株式会社, “国内標的型サイバー攻撃分析レポート 2018 年版 ～「正規」を隠れ蓑にする攻撃者～,” 2018.

[7] abuse.ch, “SSL Blacklist,”, https://sslbl.abuse.ch/, (参照 2018-08-01).

[8] M. Husák, M. Cermák, T. Jirsík , P. Celeda, “Network-Based HTTPS Client Identification Using SSL/TLS Fingerprinting,” 2015 10th International Conference on Availability, Reliability and Security, 2015.

[9] K. Stewart, “TLS Fingerprinting - a method for identifying without decrypting,” F5 Networks Inc., https://devcentral.f5.com/articles/tls-fingerprinting-a-method-for-identifying-a-tls-client-without-decrypting-24598, (参照 2018-08-01).

[10] B. Anderson , D. McGrew, “Identifying Encrypted Malware Traffic with Contextual Flow Data,” AISec '16 Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security , 2016.

[11] B. Anderson, S. Paul , D. McGrew, “Deciphering Malware's use of TLS (without Decryption),” ArXiv e-prints, 2016.

[12] F. Strasák, Detection of HTTPS Malware Traffic, Czech Technical University, 2017.

[13] The Bro Project, “The Bro Network Security Monitor,” https://www.bro.org/, (2018-08-01).

[14] R. J. Hyndman, “The problem with sturges rule for constructing histograms,” Monash University, 1995.

[15] The Stratosphere IPS Project, “Datasets Overview − Stratosphere IPS,”, https://www.stratosphereips.org/datasets-overview/, (参照 2018-08-01).

[16] “Malware-Traffic-Analysis.net,” https://www.malware-traffic-analysis.net/, (参照 2018-07-27).

[17] “Hybrid Analysis,” https://www.hybrid-analysis.com/, (参照 2018-07-27).

[18] The Stratosphere IPS Project, “Normal Captures − Stratosphere IPS,” https://www.stratosphereips.org/datasets-normal, (参照 2018-07-27).

付録

表 6 データセットに含まれるマルウェア・ファミリー

データ取得元	マルウェア・ファミリー名	データ数
CTU（Malware）	Shifu	1
	Kovter	1
	Zbot	1
	TrickBot	10
	Dridex	6
	Artemis	3
	Ursnif	1
	Cobalt	1
	CoreBot	1
	TrickStar	1
	Banking;Trojan	1
Malware-Traffic-Analysis	ZeusPanda	22
	Trickbot	17
	Dyre	3
	Kovter	1
	Ursnif V3	1
	Ursnif	1
	IcedID	2
	Zusy	1
	NanoCore RAT	1
	Generic;Trojan	2
Hybrid-Analysis	Bankshot	1
	PowerStats	1
	Threebyte	1
	TeleBot	1
	Wonder Botnet	2
	Seduploader	1
	（続く）	

	（続き）	
	VBA downloader	1
	Olympic Destroyer	2
	Ramdo	2
	StrongPity	2
	Ursnif	1
	Gootkit	1
	NexusLogger	6