SCIS 2019 2019 Symposium on
Cryptography and Information Security
Shiga, Japan, Jan. 22 - 25, 2019
The Institute of Electronics,
Information and Communication Engineers

時系列アクセスパターンのテンプレートを用いたマルウェア感染端末検知
Detecting Malware-Infected Hosts using Templates of Time-Series Access Patterns

外口 大凱 *† Taiga Hokaguchi
大下 裕一 * Yuichi Ohsita
芝原 俊樹 * ‡ Toshiki Shibahara
千葉 大紀 ‡ Daiki Chiba
秋山 満昭 ‡ Mitsuaki Akiyama
村田 正幸 * Masayuki Murata

あらまし マルウェア感染端末の検知は，サイバー攻撃を軽減するための重要な課題の 1 つである．感染端末の検知のためにテンプレートを利用した手法が提案されてきた．テンプレートは，正規表現を用いることで攻撃者による多型な攻撃を検出するよう設計されている．そして，監視対象の通信をあらかじめ観測し，テンプレートの類似度に加えて当該 URL の頻出度を考慮することで，誤検知を抑えつつ感染端末を検出していた．しかし，十分な良性通信の観測情報が得られない状況下では，多数の誤検知を生じる．本論文では，十分な良性通信の観測情報が得られない場合でも，高い検知率と低誤検知率の両立ができる手法について検討を行う．本論文では，多くのマルウェアが複数の HTTP 通信を発生させていることを考慮し，当該端末が一定時間内に発生している HTTP リクエストを束ねた，HTTP リクエスト群の特徴を捉えたテンプレートを生成することを検討する．実際のトラヒックデータを提案手法に適用した結果，閾値を制御することで従来手法に比べ，高い検知率かつ低誤検知率を達成することができた．

キーワード マルウェア, テンプレート

1 はじめに

マルウェア感染端末は，ネットワークサービスにとって大きな脅威となっている．マルウェア感染端末は，コマンドアンドコントロールサーバ (C&C サーバ) と呼ばれる攻撃者が立ち上げたサーバ経由で制御される [1, 2]．このような攻撃者により制御される感染端末はボットと呼ばれ，ボットのグループからなるネットワークはボットネットとよばれる．攻撃者は C&C サーバ経由でボットに攻撃の指示を送信することにより，サイバー攻撃を行い，その結果を C&C サーバ経由で受け取る．この攻撃では，サイバー攻撃の被害者からは，ボットからの通信しか確認できず，攻撃者自身の特定を行うことは難しい．ボットネットへの対策として，ボットネットそのものを壊滅させることや，ボットネット上のボットや C&C 通信を検出することが挙げられる [3]．ボットネットそのものを壊滅させることによってサイバー攻撃を阻止できるが，C&C サーバの構造を把握する必要があり，大規模なシステムの構築が必要である．したがって，ボットとなるネットワーク上のマルウェアに感染した端末を検出することはサイバー攻撃を軽減するための重要な課題の 1 つである．

マルウェア感染端末を検出する手法として，ホストベースの手法とネットワークベースの手法の 2 種類が挙げられる [4, 5, 6]．ホストベースの手法では、各端末上で，当該端末上でマルウェアが動作していないかの検知を行うプログラムを動作させる．しかしながら，この手法では，感染した端末が攻撃者によって制御されている場合，マルウェア検知を行うプログラム自体の動作が停止させられることも考えられる．そのため，マルウェア感染端末の検知は，ネットワークベースで行うことが効果的であり，C&C 通信に焦点を当てたマルウェア感染端末の検知手法の研究が進められてきた．

C&C 通信に焦点を当てた感染端末の検知手法として、既知の C&C サーバのドメイン名や URL と発生した通

* 大阪大学 大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5.
Graduate School of Information Science and Technology, Osaka University, Japan,1-5, Yamadaoka, Suita-shi, Osaka, 565-0871, Japan.
† E-mail: t-hokaguchi@ist.osaka-u.ac.jp
‡ 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．NTT Secure Platform Laboratories,3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan.

信を比較することにより検知を行うブラックリスト方式が考えられる [7]。この手法では，ある通信がブラックリストにあるものと一致した場合，C&C 通信と感染した端末を検出することができる．しかし，攻撃者は，自身が用いる C&C サーバがブラックリストと一致することを避けるため，C&C サーバのドメイン名や URL を頻繁に変更する [8]．そのため，ブラックリスト方式では検知できない感染端末が多く発生する．そのため，特定の URL と比較するのではなく，正規表現等を用い，攻撃者が C&C サーバに用いる HTTP リクエストの規則をテンプレートとする研究が進められている [9, 10]．これらの手法では，これまでの観測されたマルウェアが発生させた HTTP リクエストをもとに，マルウェアが発生させる通信のテンプレートを作成する．そして，監視対象の機器が通信を発生させた際に，当該通信とテンプレートの類似度をもとにしたスコアを計算し，スコアが閾値を上回った場合に，マルウェアの感染を検知する．しかしながら，マルウェアが発生させた通信をもとに生成されたテンプレートとの類似度が高い通信は，ユーザが通常行う良性な通信においても発生する．そこで，監視対象のネットワークで普段の通信で発生する HTTP リクエストの頻出度を表す希少性と呼ばれる指標が用いられる．希少性は，普段の通信を観測することにより，推定される．そして，新たに発生した通信がマルウェアによるものかを判別する際には，テンプレートとの類似度のみならず，当該 HTTP リクエストの希少性についても考慮してスコアを計算する手法を採用している．また，文献 [11] では，機械学習技術を活用してテンプレートを生成することで，C&C 通信と良性通信を分類している．これらの手法はいずれも，良性通信を事前に観測することにより誤検知を抑制している．しかし，十分な良性通信の観測結果が得られない状況においては，多数の良性通信の誤検知を生じる，あるいは，多くのマルウェア感染端末を検知できない事態に陥る．

そこで，本論文では，十分な良性通信の観測情報が得られない場合であっても，高い検知率と低い誤検知率を両立することができる手法について検討を行う．本論文では，多くのマルウェアが複数の HTTP リクエストを発生させていることを考慮し，当該端末が一定時間内に発生している HTTP リクエストを束ねた，HTTP リクエスト群の特徴を捉えたテンプレート（以降，群テンプレートと呼ぶ）を生成することを検討する．HTTP リクエスト単体でテンプレートと比較する手法と比べ，正常な通信がマルウェアが発生させている複数の HTTP リクエストの特徴に合致することは少なく，誤検知を抑えながら，マルウェアの検知率を高く維持することが期待できる．

本論文では，上記の群テンプレートをマルウェア感染端末の検知に用いる手法をプログラムとして実装し、その検知性能の評価を行った．その結果，群テンプレートを用いることにより，良性通信の観測情報を用いずとも，マルウェア感染端末の高い検知率と良性通信の低い誤検知率を両立させることができることを示す．

本論文では，2 節にて，提案手法のベースとなる，HTTP リクエスト単体に対するテンプレートを用いた感染端末検知手法である BotProfiler を説明後，3 節で群テンプレート生成と群テンプレートマッチングを取り入れた提案手法について説明する．4 節では提案手法を用いた評価結果について説明し，5 節で提案手法の評価結果に基づき考察する．6 節で提案手法の関連研究について述べ，7 節で結論を述べる．

2 BotProfiler

文献 [10] では，BotProfiler という感染端末検知手法を提案している．本節では，まず，BotProfiler の概要について説明後，テンプレートの生成手法とテンプレートマッチングの手法について説明する．

2.1 BotProfiler の概要

BotProfiler は，マルウェアが発生させている HTTP リクエストのテンプレートを生成し，監視対象の端末からの通信が観測された場合は，当該通信に含まれる HTTP リクエストと生成したテンプレートを比較することにより，マルウェアの感染端末の検知を行う．図 1 に，BotProfiler の概要を示す．BotProfiler はステップ 1 : 可変性プロファイリング，ステップ 2 : テンプレート生成，ステップ 3 : 希少性プロファイリング，ステップ 4 : テンプレートマッチングの 4 つのステップからなる．ステップ 1 と 2 は，マルウェアのサンプルを有し，サンドボックスシステム上でマルウェアを実行することができる環境で実行され，サンドボックスシステム上で観測された通信からテンプレートを生成する．ステップ 3 と 4 は，展開先ネットワークにおいて行われ，テンプレートの類似性とテンプレート内の各要素の希少性の 2 つの基準に基づき，観測された通信の悪性スコアを算出，スコアをもとにマルウェア感染端末を検知する．

2.2 テンプレートの生成

2.2.1 ステップ 1 : 可変性プロファイリング

収集されたマルウェアサンプルをサンドボックス上で実行し，発生した通信をキャプチャする．BotProfiler では，観測された通信に含まれる，HTTP リクエストに対してテンプレートを生成する．マルウェアは，既存のマルウェア等のコードを再利用して作成されることが多く，マルウェアが生成する HTTP リクエストにおいても，複数のマルウェアで共通した文字列が出現する．そ

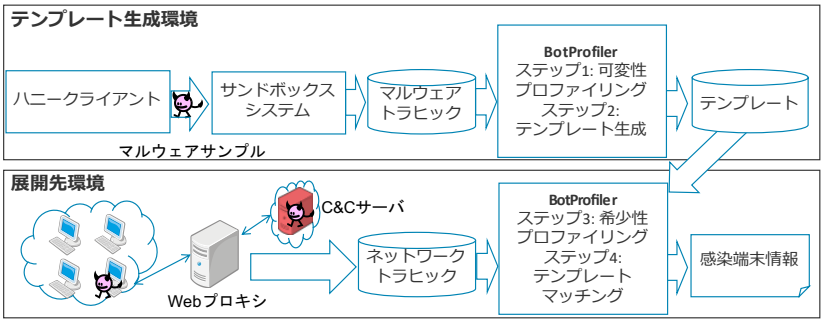

図 1: BotProfiler の概要.

表 1: 正規表現の例.

データの型	正規表現
文字列	<str; *length*>
整数値	<int; *length*>
16 進数	<hex; *length*>
64 進数	<base64; *length*>

のため，BotProfiler では，このような共通した文字列を不変キーワードとして抽出する．不変キーワードは，URL パス，URL クエリ，ユーザエージェントの 2 語以上の文字からなる文字列を候補キーワードとし，マルウェアが発する HTTP リクエストにおいて，各候補キーワードの出現頻度を求め，出現頻度が多いキーワードを不変キーワードとすることにより抽出される．

2.2.2 ステップ 2：テンプレート生成

不変キーワードを特定後，テンプレートを生成する．まず，HTTP リクエストを記号 (例えば，‘/’，‘?’，‘=’，‘-’，‘.’) を含む部分文字列に分割し，不変キーワード以外の文字列を表 1 のようにデータの型と文字数 (例えば，<str; *length*>) を含んだ正規表現に置き換える．

置換後，同様の HTTP リクエストを 1 つのテンプレートとして集約する．具体的には，HTTP リクエストを 2 段階のクラスタリングを用い，クラスタ内で代表となるものをテンプレートとして抽出する．第一段階では，宛先 IP アドレスを用いたクラスタリングを行う．同じ宛先 IP アドレス，またはプレフィックスを共有する HTTP リクエストをグループ化し，IP クラスタを生成する．第二段階では，各 IP クラスタ内で HTTP リクエスト同士の類似度によるクラスタリングを行う．本クラスタリングにおいては，次式であらわされる HTTP リクエスト h_a と h_b の類似度を用いる．

$$Sim\left(h_a, h_b\right) = \frac{1}{n} \cdot \sum_{k=1}^{n} \sigma_k\left(h_a, h_b\right) \tag{1}$$

ただし，σ_k は HTPP リクエスト h_a と h_b の要素間の類似度を計算する関数であり，n は要素の数を表す．σ_1 は，URL パス間の類似度であり，正規化された編集距離を用いて計算される．σ_2 は URL クエリにおけるパラメータ名の組み合わせの類似度であり，Jaccard 類似度を用いて計算される．σ_3 は，URL クエリにおける値の類似度であり，同じデータの型と長さのものの比率を用いて計算される．σ_4 はユーザエージェント間の類似度であり，正規化された編集距離を用いて計算される．また，上記の類似度は $\sigma_k \in [0,1], \forall k$ かつ $Sim\left(h_a, h_b\right) \in [0,1]$ である．

類似クラスタ生成後，各クラスタにおいて重心となる HTTP リクエストを抽出する．すなわち，ある HTTP リクエストと他の全てのリクエストとの類似度の合計が最大となる HTTP リクエストを重心として抽出する．重心として抽出された HTTP リクエストがテンプレートとして使用される．

2.3 テンプレートマッチング

2.3.1 ステップ 3：希少性プロファイリング

ステップ 3 では，ステップ 2 で生成されたテンプレートを構成する要素のうち，展開先において，稀にしか観測されない要素を発見する．BotProfiler では，URL パス，URL クエリ，ユーザエージェントの希少性に注目している．テンプレート t 中の要素 k の希少性 $\rho_{t,k}$ は，以下の式から算出される．

$$\rho_{t,k} = 1 - \frac{n_{t,k}}{max_i n_i} \tag{2}$$

$n_{t,k}$ は t で k を含む HTTP リクエストを送信しているホスト数であり，$max_i n_i$ は最大ホスト数を表す．上記の式において，$\rho_{t,k} \in [0,1], \forall t, \forall k$ である．

2.3.2 ステップ 4：テンプレートマッチング

ステップ 4 では，ステップ 2 で生成されたテンプレートステップ 3 で生成された希少性の 2 つと評価するトラヒックを比較し，マルウェアに感染したホストを検出する．具体的には，HTTP リクエスト h とテンプレー

ト t との類似度をスコア $Score(h,t)$ を以下の式から算出する．

$$Score(h,t) = \frac{\sum_{k=1}^{n} \sigma_k(h,t) \cdot \omega(\sigma_k(h,t), \rho_{t,k})}{\sum_{k=1}^{n} \omega(\sigma_k(h,t), \rho_{t,k})} \cdot \rho_{h,d} \quad (3)$$

$\sigma_k(h,t)$ は，類似クラスタを生成した際に使用したものと同様の類似度を使用し，$\rho_{t,k}$ は t における k の希少性を表す．$\rho_{h,d}$ は完全修飾ドメイン名 (FQDN) の希少性であり，ステップ 3 と同様に計算される．ω は $\sigma_k(h,t)$ と $\rho_{t,k}$ の重みであり，次式のように定義される．

$$\omega(\sigma_k(h,t), \rho_{t,k}) = 1 + \frac{1}{(2 - \sigma_k(h,t) \cdot \rho_{t,k})^m} \quad (4)$$

ここで m はパラメータを表す．$Score(h,t) \in [0,1]$ であり，テンプレート t との類似度が高く，展開先ネットワークにおいて t の要素の希少性が高い場合，$Score(h,t)$ は高くなるように設計される．$Score(h,t)$ があらかじめ定義された閾値 θ_H を超えた場合，評価する HTTP リクエストがテンプレートに一致しているとみなされ，マルウェアに感染したホストとして検出する．

3 提案手法

BotProfiler では，テンプレートの一致度とテンプレート内の要素の希少性を用いることで感染端末を検知していた．しかしながら，展開先ネットワークの良性通信を十分に観測できていない状況下においては，希少性の見積もりが展開先ネットワークの状況に合わず，誤検知を生じる，あるいは，誤検知を避けるように閾値を設定すると，検知ができないマルウェア感染端末を生じてしまう．

そこで，本論文では，良性通信が十分に観測できていない状況下においても，高い検知率と低い誤検知率を両立することができる手法を検討する．BotProfiler [10] の手法で生成されたテンプレート (以降，単体テンプレートと呼ぶ) との一致度が非常に高い通信は，良性な通信であることは稀であり，逆に，一致度が非常に低い通信は悪性であることは稀である．そのため，本論文では，このような従来手法により，検知できる通信は除外し，従来手法では検知が困難であった，単体テンプレートとの一致度が中程度となった通信に焦点を当てる。そして，本論文では，多くのマルウェアが複数の HTTP リクエストを発生させていることを考慮し，単体テンプレートとの類似度が中程度となった通信については、当該端末が一定時間内に発生している HTTP リクエストを束ねた，HTTP リクエスト群の特徴を捉えた群テンプレートを生成する．HTTP リクエスト単体で単体テンプレートと比較する手法と比べ，正常な通信が，マルウェアが発生させている複数の HTTP リクエストの特徴に合致することは少なく，誤検知を抑えながら，マルウェアの検知率を高く維持することが期待できる．

3.1 提案手法の概要

提案手法の概要を図 2 に示す．提案手法では，BotProfiler で生成した単体テンプレートに加え，複数の通信の特徴をとらえた群テンプレートを保持する．群テンプレートは，マルウェアが一定時間以内に生成する通信をテンプレート化したもので、各群テンプレート T は，1 つの感染端末が発生させている HTTP リクエストに 1 回以上 BotProfiler でマッチした単体テンプレートの集合 U_T と，U_T の各単体テンプレートにマッチした回数 $t_i\,(i \in U_T)$ からなる．

以下に，群テンプレートの生成方法と，BotProfiler で生成した単体テンプレート，群テンプレートの両方を用いた提案手法におけるマルウェア感染端末検知の手順について述べる．

3.2 群テンプレートの生成

提案手法では，BotProfiler にて単体テンプレート生成後，単体テンプレート生成に用いたマルウェアトラヒックを再度用いて群テンプレートを生成する．具体的には，各マルウェアが送出する HTTP リクエストと単体テンプレートを照合し，類似度が最も高い単体テンプレートとの対応づけを行う．その後，マルウェアごとにマッチした単体テンプレートとマッチ回数を集約し，群テンプレートとして生成する．マルウェアごとに最も類似度が高い単体テンプレートとマッチ回数を集約することで，複数の通信を行う感染端末の挙動を捉えることができる．

3.3 検知手順

提案手法では，監視対象の各端末が発する HTTP リクエストを時間で分割し，HTTP リクエストの群を構築し，その群に悪性な通信が含まれていないか，群テンプレートとのマッチングにより検知を行う．この群テンプレートとのマッチングは，以下の手順により行う．

3.3.1 単体テンプレートとのマッチング

HTTP リクエスト群に含まれる各 HTTP リクエストについて，BotProfiler が生成した単体テンプレートとのマッチングを行う．ただし，本論文では，希少性を算出するのに十分な正常な通信の観測が得られない状況を考えているため，希少性は用いない．すなわち、HTTP リクエスト h とテンプレート t の類似度スコア $Score(h,t)$ は以下の式で算出する．

$$Score(h,t) = \sum_{k=1}^{n} \sigma_k \quad (5)$$

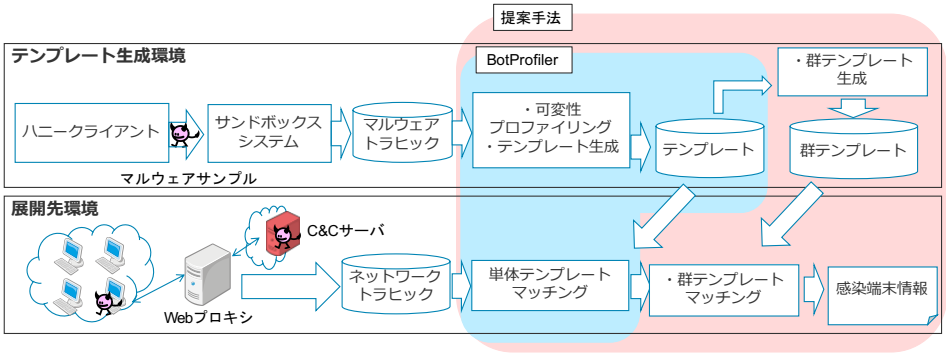

図 2: 提案手法の概要.

その後，HTTP リクエスト群中のすべての HTTP リクエストが、いずれの単体テンプレートに対しても $Score(h,t)$ が閾値 θ_L 以下であれば，当該 HTTP リクエスト群は良性であると判断し，HTTP リクエスト群中に $Score(h,t)$ が閾値 θ_H 以上となる単体テンプレートが存在するリクエストがあれば，当該 HTTP リクエスト群は悪性であると判断する．上記のいずれの場合でもない場合は，以下の群テンプレートマッチングを行う．

3.3.2 群テンプレートマッチング

群テンプレートマッチングでは，HTTP リクエストの群と群テンプレートのマッチングを行う．群テンプレートとのマッチングは，群テンプレートに含まれる単体テンプレートとマッチする HTTP リクエストが，HTTP リクエストの群にどれだけ含まれているかという点を反映した群スコアをもとに行う．群スコアについては，様々な定義の仕方が考えられるが，本論文では，群テンプレートに含まれる単体テンプレートのうち，HTTP リクエスト群中にマッチした単体テンプレートの割合が高くなれば，群スコアの値も高くなるように，以下の式で定めた $S(D,T)$ を用いた．

$$S(D,T) = 1 - \frac{1}{|U_T|} \sum_{i \in U_T} s(d_i, t_i) \tag{6}$$

$$s(d_i,t_i) = \begin{cases} \alpha & (d_i = 0) \\ \frac{\beta(t_i - d_i)}{t_i} & (0 < d_i \leq t_i) \\ 0 & (d_i > t_i) \end{cases} \tag{7}$$

ただし，D は HTTP リクエスト群，T は群テンプレートであり，d_i は，単体テンプレート i との類似度スコア $Score(h,t)$ が閾値 θ_L を超えた D 中の HTTP リクエストの数を示す．また，α, β はそれぞれパラメータであり，$\alpha >> \beta$ である．

群スコアを計算し，群スコアの閾値が θ_G 以上であった場合に，当該 HTTP リクエスト群は悪性であると検知する．このように群スコアを計算することにより，各 HTTP リクエストの単体テンプレートとの類似度が θ_H 以上とならずに，HTTP リクエスト単体では悪性であると検知することが難しい場合であっても，HTTP リクエストを束ねることにより，マルウェアが生成する通信の挙動を確認することができ，マルウェア感染端末の検知を行うことができる．

4 評価

4.1 評価環境

データセット 本評価では，MDL [12]，hpHosts [13] および商用ブラックリストといったパブリックブラックリストをクローリングして収集したマルウェアサンプルをサンドボックスシステム [14] で実行することにより収集した，マルウェア感染端末が発する通信を用いる．なお，本サンドボックスは，Microsoft Windows 環境で実行可能なファイルのみをサポートしている．表 2 に示すように，本評価ではマルウェアサンプルを収集された時期によって教師データとテストデータに分割した．

本評価では，良性データはユーザに同意の元，実際に大学内で日常的なネットワーク利用の際に生じているトラヒックを収集することにより得た．本評価では，良性通信の観測情報が得られない状況を想定しているため，良性の HTTP リクエストはテストデータのみに使用する．

HTTP リクエスト群の生成 提案手法では，ある HTTP リクエストからあらかじめ設定した時間 s 秒以内に送信されている HTTP リクエストを 1 つの HTTP リクエスト群として生成する．時間 s は提案手法におけるパラメータであり，s の値を適切に設定しないと，マルウェアの挙動を捉えた HTTP リクエスト群がされない可能性がある．

そこで，表 2 のテストデータに含まれるマルウェアが生成する HTTP リクエストを調べ，感染端末が通信を開始してから終了するまでの時間を調べた．図 3 に，感染端末の通信時間と端末数の分布を示す．図 3 には，全テストデータにおける分布のみではなく，単体テンプレートとのマッチング時のスコアが 0.8 を超えた HTTP リ

表 2: データセット．

ラベル	教師		テスト	
	期間	HTTP リクエスト数	期間	HTTP リクエスト数
悪性	2017/8/1 - 2017/12/31	656,714	2018/1/1 - 2018/3/31	442,532
良性	-	-	2018/3/1 - 2018/3/31	293,120

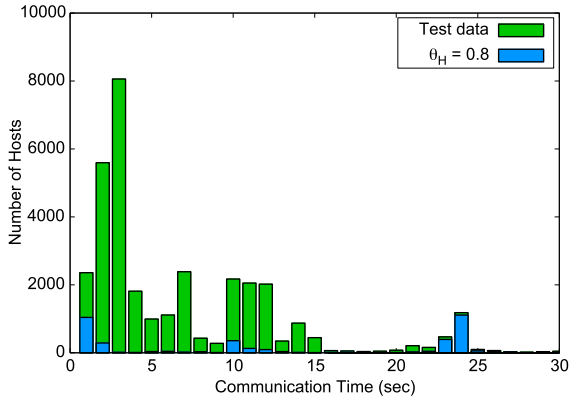

図 3: 感染端末の通信時間の端末数の分布．

クエストが存在する端末以外の感染端末における通信時間の分布を示している．また，図 3 において，全テストデータにおいて通信時間が 30 秒以上である端末数は極めて少ないため，省略する．

全テストデータの分布をみると，通信時間が 3 秒の感染端末が最も多い．しかしながら，単体テンプレートとのマッチングでは検知できない，単体テンプレートとのマッチングスコアが 0.8 未満の感染端末に注目すると，通信時間が 1 秒である端末と 24 秒である感染端末が多いことがわかる．そこで，本論文では，通信時間が 24 秒ある感染端末の特徴もとらえることができるように，$s = 24$ とし，HTTP リクエスト群を生成した．本評価では，簡単のために HTTP リクエスト群を s 秒ごとに分割し，提案手法の検知性能について調査した．

4.2 評価結果

提案手法の検知性能 提案手法にて単体テンプレートと群テンプレートを生成後，テストデータを用いて検知性能を評価する．本論文における検知率 (TPR: True Positive Rate) は，テストデータ中の全感染端末のうち，検出された感染端末の割合を示す．誤検知率 (FPR: False Positive Rate) は，生成された HTTP リクエスト群のうち，誤って検出された HTTP リクエスト群の割合を示す．表 3 に，テストデータ中の感染端末数と良性通信で生成された HTTP リクエスト群数を示す．

図 4 に，提案手法にて θ_H, θ_L をあらかじめ 5 パターン設定し，θ_G を 0.01 ずつ変化させたときの検知率・誤検知率の関係を示す．また，単体テンプレートで式 (5) の

表 3: テストデータ中の感染端末数と良性通信で生成された HTTP リクエスト群数．

感染端末数	良性 HTTP リクエスト群数
44,116	10,440

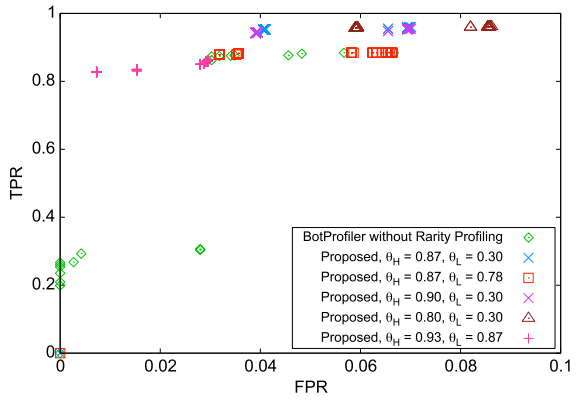

図 4: BotProfiler without RP と提案手法おける検知率・誤検知率の関係．

スコアに対して θ_H を超えた HTTP リクエストが存在すると，感染端末として検知する手法（以降，BotProfiler without RP: Rarity Profiling と呼ぶ) の検知率・誤検知率の関係を示す．BotProfiler without RP についても，θ_H を 0.01 ずつ変化させて，検知率・誤検知率を調べた．

図 4 において，横軸は誤検知率，縦軸は検知率を表す．閾値 $\theta_L, \theta_H, \theta_G$ を制御することで，提案手法は，BotProfiler without RP と比べて検知率が高く，誤検知率が低い点が存在していることがわかる．

表 4 は，$\theta_H = 0.93, \theta_L = 0.87$ として θ_G を変えたときの提案手法の検知率・誤検知率について示す．θ_G を 0.75 から 0.90 まで変化させても検知率・誤検知率に大きな変化はなく，高い検知率かつ低誤検知率を達成できる．これは，群テンプレートのマッチングにより，複数の単体テンプレートとマッチさせることが，誤検知率を低く抑えつつ，高い検知率を維持するのに有効であるためである．群スコアが 0.75 以上になるのは，群テンプレート内の複数の単体テンプレートと一致するときのみであり，$\theta_G \geq 0.75$ とすることにより，複数の単体テンプレートと一致した場合のみを感染端末として検知することができる．

表 4: θ_G を変えたときの提案手法の検知率・誤検知率の比較.

θ_G		0.90	0.85	0.80	0.75
Proposed	TPR	82.76%	82.77%	82.83%	83.02%
	FPR	0.88%	0.88%	0.88%	1.64%

表 5: BotProfiler without RP の検知率と，θ_L,θ_H を変えた際の提案手法の検知率.

θ_H			0.87	0.90	0.93
TPR	BotProfiler without RP		87.42%	30.48%	29.28%
	Proposed	θ_L			
		0.3	95.23%	94.35%	92.94%
		0.78	87.87%	86.12%	84.69%
		0.87	-	84.23%	82.77%
FPR	BotProfiler without RP		3.02%	2.79%	0.42%
	Proposed	θ_L			
		0.3	4.06%	3.90%	2.03%
		0.78	3.18%	3.02%	1.14%
		0.87	-	2.75%	0.88%

最後に, BotProfiler without RP と, 提案手法で θ_L,θ_H を変えたときの検知性能の比較を表 5 に示す．ここでは，$\theta_G = 0.85$ とする．θ_H を大きくすると検知率は低下するが，誤検知率は小さくなる．同様に，θ_L を大きくすると検知率は低下するが，誤検知率は小さくなる．また，θ_H を高い値に設定しつつ，θ_L を θ_H よりも低い値に設定することにより，低い誤検知率を維持したまま，高い検知率を達成することができる．これは，単体テンプレートマッチングにて検知漏れとなった悪性通信が，群テンプレートマッチングによって検知されるからであると考えられる．

5 考察

良性通信と悪性通信の混在環境への適用　本論文では，悪性通信と良性通信に対してそれぞれ独立に HTTP リクエスト群を生成し，提案手法に適用した．しかし，実際には悪性通信と良性通信が混在しており，混在した環境下で HTTP リクエスト群を生成する必要がある．提案手法における群テンプレートマッチングでは，悪性通信がマッチした単体テンプレートに対して θ_G が高くなるよう設計されているため，HTTP リクエスト群に悪性通信と良性通信が混ざっている場合でも，良性通信の影響を受けずに θ_G が高くなり，感染端末を検知できると考えられる．

HTTP リクエスト群の生成手法　前節の評価においては, 評価を簡単にするため, 正常な HTTP リクエストを s 秒ごとに区切ることにより，正常な通信における HTTP リクエスト群を生成した．しかしながら，感染端末を検知するためには，感染端末が発する特徴的な通信をすべて含む HTTP リクエスト群が生成されることが望ましいが，発生する通信を s 秒ごとに区切る方法では，感染端末が発する特徴的な通信が二つの群に分割されてしまうことも考えられる．この問題に対しては，s 秒の長さのウィンドウをスライディングさせながら，HTTP リクエスト群を生成することが有効であると考えられる．

パラメータ設定　本論文では，提案手法による評価の結果，適切なパラメータを与えることで従来手法より検知性能が向上することを示した．これらのパラメータを自動的に適切な値に設定することは，今後の課題であり，今後，悪性な通信データのみからパラメータを適切に設定する手法を考案し，評価を行う予定である．

6 関連研究

マルウェア感染端末を検知する様々な手法が提案されてきた．文献 [4] は，マルウェア感染時に発生する共通のアクションの関係をモデル化し，感染の検出を行う手法を提案している．ただし，本手法はホストベースの手法であり，ホストベースの手法では，当該端末が攻撃者により制御される場合，検知を行うプロセス自体が攻撃者によって停止され，感染の検知を行うことができないといった問題も生じうる．

そのため，ネットワークベースの検知手法も広く検討されている．文献 [5] は，C&C 通信で使用される URL を検出する，正規表現を用いたシグネチャを生成する手法を提案しており，文献 [9] では，マルウェアが発する HTTP 通信に対して正規表現を用いたテンプレートを生成し，テンプレートの類似性だけでなく，希少性を用いて感染端末を検出する．文献 [10] は文献 [9] を拡張したものであり，攻撃者が既存のマルウェア等のコードを再利用して作成することが多く，マルウェアが生成する HTTP リクエストにおいても，複数のマルウェアで共通の文字列が出現するといった点に焦点を当てている．このような共通した文字列を不変キーワードとして抽出し，不変キーワード以外の文字列を正規表現に置き換えたテンプレートを生成する．さらに，文献 [11] は機械学習技術を活用し，トラヒックを悪性と良性に分類する．分類後，類似した特徴を集約し，テンプレートを生成する．しかしながら，これらの研究においては，正確な悪性通信の検知のためには，良性通信の観測情報を用いて希少性の計算を行うことや，機械学習を行うことが必要であり，十分な良性通信の観測情報が得られない状況下では多くの誤検知を発生する，あるいは検知漏れが発生する．提案手法は，複数の通信に着目し，良性通信の観測情報がない状況下においても高い検知率を低い誤検知率を両立できる．この問題に対して，本論文では，単一

の HTTP 通信に対するテンプレートとのマッチングを行うのみではなく、一定時間以内に発生した HTTP 通信を束ねた HTTP 通信群について，その特徴をとらえたテンプレートを生成し、マッチングを行うことにより，良性通信の観測情報がない状況下であっても，高い検知率と低い誤検知率の両立を狙うものである．

複数の通信を考慮した悪性検知手法としては，ドライブバイダウンロード攻撃を対象として我々が提案した手法 [15] が挙げられる．ドライブバイダウンロード攻撃は，攻撃者が改ざんサイトから多段のリダイレクトを通して，マルウェアダウンロードサイトまで誘導することにより行われ，本手法は，そのような多段のリダイレクト構造をドライブバイダウンロード攻撃の特徴として、学習を行う。学習には，ドライブバイダウンロード攻撃に含まれる HTTP リクエストの系列と，日常的に発生する良性な HTTP リクエストの系列を用いる．それに対して本論文では，マルウェアが発生させる各 HTTP リクエストについて，当該 HTTP リクエストに該当するリクエストが存在しているかに注目して群テンプレートを生成・検知に用いる手法を提案している．これにより，良性通信の観測情報が存在しない場合であっても，感染端末を検知するのに有用なテンプレートの生成が可能となっている．

7 おわりに

本論文では，十分な良性通信の観測情報が得られない場合でも，高い検知率と低誤検知率の両立ができる手法について検討を行った．本論文では，多くのマルウェアが複数の HTTP 通信を発生させていることを考慮し，当該端末が一定時間内に発生している HTTP リクエストを束ねた，HTTP リクエスト群の特徴を捉えたテンプレートを生成することを検討した．実際のトラヒックデータを提案手法に適用した結果，閾値を制御することで従来手法に比べ，高い検知率かつ低誤検知率を達成することができた．今後の課題として，HTTP リクエスト群を生成する際の適切な時間の設定手法，また，閾値の調整手法が挙げられる．

参考文献

[1] M. Feily, A. Shahrestani, and S. Ramadass, “A survey of botnet and botnet detection,” in *Proceedings of Third International Conference on Emerging Security Information, Systems and Technologies*. IEEE, 2009, pp. 268–273.

[2] A. Emigh, “The crimeware landscape: Malware, phishing, identity theft and beyond,” *Journal of Digital Forensic Practice*, vol. 1, no. 3, pp. 245–260, 2006.

[3] R. Boscovich, “Microsoft and financial services industry leaders target cybercriminal operations from zeus botnets,” *The official Microsoft blog*, 2012.

[4] G. Gu, P. A. Porras, V. Yegneswaran, M. W. Fong, and W. Lee, “Bothunter: Detecting malware infection through ids-driven dialog correlation.” in *Proceedings of USENIX Security Symposium*, vol. 7, 2007, pp. 1–16.

[5] R. Perdisci, W. Lee, and N. Feamster, “Behavioral clustering of http-based malware and signature generation using malicious network traces.” in *Proceedings of 15th USENIX Symposium on Networked Systems Design and Implementation*, vol. 10, 2010, p. 14.

[6] R. Bortolameotti, T. van Ede, M. Caselli, M. H. Everts, P. Hartel, R. Hofstede, W. Jonker, and A. Peter, “Decanter: Detection of anomalous outbound http traffic by passive application fingerprinting,” in *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 2017, pp. 373–386.

[7] M. Kührer, C. Rossow, and T. Holz, “Paint it black: Evaluating the effectiveness of malware blacklists,” in *Proceedings of International Workshop on Recent Advances in Intrusion Detection*. Springer, 2014, pp. 1–21.

[8] P. Vinod, R. Jaipur, V. Laxmi, and M. Gaur, “Survey on malware detection methods,” in *Proceedings of the 3rd Hackers’ Workshop on computer and internet security*, 2009, pp. 74–79.

[9] T. Nelms, R. Perdisci, and M. Ahamad, “Execscent: Mining for new c&c domains in live networks with adaptive control protocol templates.” in *Proceedings of USENIX Security Symposium*, 2013, pp. 589–604.

[10] D. Chiba, T. Yagi, M. Akiyama, K. Aoki, T. Hariu, and S. Goto, “Botprofiler: Detecting malware-infected hosts by profiling variability of malicious infrastructure,” *IEICE Transactions on Communications*, vol. 99, no. 5, pp. 1012–1023, 2016.

[11] S. Mizuno, M. Hatada, T. Mori, and S. Goto, “Botdetector: A robust and scalable approach toward detecting malware-infected devices,” in *Proceedings of IEEE International Conference on Communications*. IEEE, 2017, pp. 1–7.

[12] “Malware domain list.” http://www.malwaredomainlist.com/, accessed December 10. 2018.

[13] “Malwarebytes,” http://www.hosts-file.net/, accessed December 10. 2018.

[14] K. Aoki, T. Yagi, M. Iwamura, and M. Itoh, “Controlling malware http communications in dynamic analysis system using search engine,” in *Proceedings of Third International Workshop on Cyberspace Safety and Security*. IEEE, 2011, pp. 1–6.

[15] T. Shibahara, K. Yamanishi, Y. Takata, D. Chiba, T. Hokaguchi, M. Akiyama, T. Yagi, Y. Ohsita, and M. Murata, “Event de-noising convolutional neural network for detecting malicious url sequences from proxy logs,” *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 101, no. 12, pp. 2149–2161, 2018.