サイバー攻撃観測における対象ネットワーク特化型ホワイトリスト作成手法の提案

金谷 延幸[1,a)]　津田 侑[1]　高野 祐輝[1,2]　井上 大介[1]

概要：
セキュリティインシデントの監視・対応においては，大量に発生する通信ログやセキュリティ機器のアラート情報などの観測データを目視で確認する必要がある．現実の運用では，着目すべき観測データをフィルタリングするため，より頻繁かつ定常的に発生する通信をホワイトリストとして規定することが多い．従来のホワイトリストは，Alexa ランキングなどを参考に観測者の経験に基づき設定されていた． しかし，この方法では設定者の経験への依存性の高さや経年変化への対応の困難さに課題があった．そこで我々は，組織内ネットワークで収集された通信データから特徴量の発生分布を求め，その標準偏差を通信の定常性に対する指標とすることで，ホワイトリストを自動生成する手法を提案する．大量の通信データに対する特徴量を発生分布に畳み込み標準偏差によって数値化することで，対象ネットワークの特徴を細かく反映した単純な数値指標を得ることができる．この手法を実装し，異なる２つのネットワーク環境に対し適用することで評価を行い，その有効性を確認できた．

キーワード：ホワイトリスト，IDS，インシデント対応，マルウェア解析

A Generation Method for Whitelist to fit Own Network in Cyber Attack Monitoring

Nobuyuki Kanaya[1,a)]　Yu Tsuda[1]　Yuuki Takano[1,2]　Daisuke Inoue[1]

Abstract: In monitoring and responding to security incidents, it is necessary to check data such as a large amount of logs and alert information of security devices. In actual operation, in order to filter the observation data to be noticed, communication that occurs more frequently and regularly is often specified as a white list. The conventional whitelist setting was based on the experience of the observer with reference to Alexa ranking. However, this method has problems such as high dependence on the experience of the operator and difficulty in dealing with aging. Therefore, we calculate the occurrence distribution of features from communication data collected in the network, and the standard deviation. We propose a method of automatically generating a whitelist by using as an index for stationarity of communication. By convolving the feature values for large amounts of communication data with the occurrence distribution and quantifying them with the standard deviation, A simple numerical index reflecting the characteristics of the target network can be obtained. We implemented this method, applied it to two different network environments, and confirmed its effectiveness.

Keywords: whitelist, IDS, incident response, malware analysis

1. はじめに

セキュリティインシデントの監視・対応では，対象となる組織が保有するネットワークで収集されるセキュリティ

[1] 国立研究開発法人 情報通信研究機構
National Institute of Information and Communications Technology
[2] 大阪大学 大学院工学研究科
Graduate School of Engineering, Osaka University
[a)] kanaya@nict.go.jp

機器が発生させた大量のアラートや通信ログなどの大量の観測データから，必要な情報を発見し確認することが必要となる．Verwoerd らは IDS を構成する技術の一要素としてホワイトリストを挙げている [1]．Verwoerd らは，ホワイトリストを「既知の良性パターン」と定義し，その目的を大量のアラート列から既知の良性なアラートを除去し，未知の危険なアラート列を得ることでデータ量を削減することであると述べている．同時に，除去率の向上と誤検知率（ホワイトリストの場合は誤除去の可能性）の低減に必要となるホワイトリスト逐次改善の重要性とその困難性について言及している．現実の運用では，除外対象となる観測データを指定するホワイトリストを運用者が作成し，観測データをフィルタリングすることで，より着目すべき情報に絞る運用が行われる．

この除去可能な観測データを規定するホワイトリストを作成する際に参考される情報の一つとして，Alexa Top 500 [2] がある．Alexa の統計データにより上位に位置付けらるウェブサイトを安全なホストとみなしホワイトリストの要素とする．この方法は，不特定のネットワークに対し共通のホワイトリストを適用する場合に用いられる．

一方で，ある企業や組織が所有するネットワークにおける運用においては，通信ログやアラートに環境固有の特徴が現れ，共通のホワイトリストでは十分な削減率が達成できない．そこで対象ネットワークに特化したホワイトリストの作成・維持により，削減率の向上と誤検知の低減が要求される．対象ネットワークに特化したホワイトリスト作成手法として一定期間サンプリングされた通信データから単純に通信先と通信頻度に基づきホワイトリストを構成する方法がある．工業用のネットワークなど定常的な通信のみが発生するネットワークでは，サンプリングによるホワイトリスト作成手法は有効に作用する．しかしながら，一般的なオフィス等で利用されるネットワークでは非日常的な通信や，突発的な通信が常時発生する可能性があり，単純な通信量や接続数のみに依存したホワイトリストは誤検知の可能性が生じる．

そこで我々は，対象ネットワークから集計されたホスト毎の特徴量における変動に着目した定常的通信の識別に基づく，ネットワーク特化型ホワイトリスト作成手法を提案する．この手法では，組織内ネットワークで収集された通信データから，ホスト毎の特徴量の発生分布を求め，その標準偏差をホストに対する通信の定常性指標とする．大量の通信データに対する特徴量を発生分布に畳み込み標準偏差によって数値化することで，対象ネットワークの構成を反映したホスト毎の特性に対する単純な数値指標を得る．この数値指標に基づき，ホワイトリストを構成することで，高削減率と低誤検知率を両立させる．

本論文の構成は，まず 2 章にて関連研究について述べたあと，3 章にて提案手法の詳細を述べる．次に，4 章にて実装と性能評価について述べたあと，5 章にて当機構のネットワークおよび能動的観測環境のネットワークより収集した通信データを用いたケーススタディについて述べ 6 章で考察し，7 章でまとめについて述べる．

2. 関連研究

ホワイトリストを利用した異常検知に関する既存研究について，ここでは対象となるネットワークの性質によって大別して述べる．

まず，産業制御システムネットワークでのホワイトリスト手法について述べる．Choi らは SCADA のネットワークにおいて 5-tuple のフロー情報を利用したホワイトリストの生成手法を提案し 7 件の SCADA において性能を評価している [3]．Jang らは SCADA のようなクローズドで静的かつ固有のネットワークに対して有効なホワイトリストベースの侵入検知手法を提案している [4]．これらの手法では，一定期間の通信をサンプリングし，この期間に発生した通信を安全とみなし，通信先やポート番号から構成されるホワイトリストを作成する．産業制御ネットワークは，役割が固定された特定マシン間における画一的なソフトウェアによる機械的な限定された通信がほとんどであり，またその構成はほとんど変更されない．このような環境で適用されるホワイトリストは，一般的なネットワークと違い，より単純でかつ長期的に適用でき，安定的な通信を前提とした基づくホワイトリスト作成手法が有効に機能する．一方で，我々が対象とする企業ネットワークには，多様な機器やソフトウェアとその複雑な構成を対象とし，人間の不確定な行動を起点とする通信を考慮した，日々の変化に追従できるホワイトリスト作成手法が必要となる．

次に，企業ネットワークにおけるホワイトリストと作成手法について述べる．Takemori らはホスト毎に発生させる宛先 IP アドレスやドメイン名からホワイトリストを作成し異常検知に利用する手法を提案している [5]．Yen らは企業ネットワーク内で発生する HTTP(S) プロキシログを自動でマイニングする手法として Beehive を提案している [6]．Yen らは，対象とするデータ数を削減するために，アクセス数が多いホストや，「favicon」などのブラウザが機械的にアクセスするリソースをホワイトリスト化している．さらに，HTTP リクエストの Referer に着目し，安全なホストからのリンクによる通信を安全と見なし，Referer に頻出するサーチエンジンやニュースサイトの URI をホワイトリスト化し，Referer に含む場合にログを除去する．より出現頻度の低いホストの URI を含む場合や Referer の存在しない場合を危険と見なし，これらのログを解析対象とする．また Hwang らはブラックリスト-ホワイトリスト-SVM (Support Vector Machine) で構成される 3 層アーキテクチャの IDS を提案している [7]．これらの研究では経験則や，過去の出現履歴，発生頻度に基づきホワイトリ

ストを作成している．我々の手法は，通信データに対する複数の定量的指標に基づきホワイトリストを作成する点において既存研究に対し優位であると考える．

3. 発生分布に基づくホワイトとリスト生成

我々の手法では標準偏差を使いホストにおける定常通信の発生を評価するが，単純に特徴量に対する平均と標準偏差を求めるのではなく，期待される発生分布と実際の通信データから得られた発生分布の差に対する標準偏差を求めることを特徴とする．大量の通信データに対する特徴量を発生分布に畳み込み標準偏差を計算し指標とすることで，ホストにおける定常通信の特性を単純な数値として表現し比較することができる．

本章では，我々の提案する発生分布に基づく標準偏差の計算方法と，これを用いたホワイトリストの生成手法について述べる．

3.1 発生分布に対する標準偏差の計算

我々の手法では，一定の収集期間毎に区切り，収集された通信データの特徴量を求め，その特徴量をもとに標本の振り分けを行い通信データの発生分布を求る．次に全収集期間の通信データから発生確率を求め，発生確率から期待発生分布を求める．その後，各収集期間における発生分布と期待発生分布との差の自乗を求め，それらを積算し標準偏差を求める．

ここで，ある通信データ x は，その通信の通信元もしくは通信先によって決まるホスト β_i と呼ぶ値と，収集時刻 t_x という属性情報をもち，その通信データからはある特徴量 C_x を得ることができるとする．このとき，収集期間，標本区間を以下のように定義する．

収集期間 J_j とは 2 次元ベクトルであり $J_j = [J_1^j, J_2^j]$ で表される．すなわち，x の収集時刻 t_x が $J_1^j \leq t_x < J_2^j$ を満たすとき，x は収集期間 J_j に属すると定義する．

標本区間 S_l とは r 個の述語の集合であり，$S_l = \{S_1^l, \ldots S_r^l\}$ で表される．すなわち，$\exists s(s(C_x) \wedge s \in S_l)$ を充足するとき，特徴量 C_x は標本区間 S_l に属すると定義する．

すると，ある通信データの集合 $X = \{X_1, \ldots, X_n\}$ が得られたら，あるホスト β_i，収集期間 J_j，標本区間 S_l の通信発生分布 R_{ijl} は以下のように定義できる．

$$R_{ijl} = \sum_{k=1}^{n} (f(C_{X_k}, l) \cdot g(X_k, j) \cdot h(X_k, i))$$

$$f(T, l) = \begin{cases} 1 & \exists s(s(T) \wedge s \in S_l) \\ 0 & \text{otherwise} \end{cases}$$

$$g(x, j) = \begin{cases} 1 & J_1^j \leq t_X < J_2^j \\ 0 & \text{otherwise} \end{cases}$$

$$h(x, i) = \begin{cases} 1 & \text{Data } X\text{'s site is } \beta_i \\ 0 & \text{otherwise} \end{cases}$$

関数 $f(T, l)$ は特徴量 T が標本区間 S_l に属するかを判定する関数であり，関数 $g(x, i)$ と $h(x, i)$ は，それぞれ，通信データ x が収集期間 J_j に属するか，および β_i に関する通信かを判定する関数である．

発生分布の期待値である期待発生分布を E_{ijl} とすると，上記に示した発生分布 R_{ijl} を使いホスト β_i に対する標準偏差 D_i を以下の式で求める．

$$D_i = \frac{\sqrt{\sum_{j=1}^{m} \sum_{l=1}^{n} (E_{ijl} - R_{ijl})^2}}{mn}$$

すなわち全収集期間 $[1, m]$ における全発生分布 $[1, l]$ の期待値との差の自乗を求め，その和の平方根を数で割ることで，発生分布の標準偏差を求める事を上記の式は示す．

次に，あるホスト β_i における標本区間 S_l での発生確率 P_{il} を以下の式により計算する．

$$P_{il} = \frac{\sum_{j=1}^{m} R_{ijl}}{\sum_{j=1}^{m} \sum_{l=1}^{o} R_{ijl}}$$

これは，全収集期間 $[1, m]$ に対して同じ発生確率とし，全収集期間 $[1, m]$ の発生分布の平均値から求めることを表す．このとき，あるホスト β_i に対し通信が n 回発生した場合，ある標本区間 S_l に属す通信回数の期待発生分布の値はそれらの積 $n \cdot P_{il}$ 回となる．

期待発生分布 E_{ijl} は全収集期間 $[1, m]$ において同じ発生確率 P_{il} で各標本区間への振り分けが発生するとし，以下の式で表される．

$$E_{ijl} = P_{il} \sum_{l=1}^{o} R_{ijl}$$

これは，期待発生分布 E_{ijl} を，収集期間 j におけるホスト β_i に対する通信総数と各 S_l における発生確率 P_{il} の積で求めることを表す．

つまり，標準偏差 D_i は，ある特徴量と標本区間に着目した場合の，全収集期間において同じ発生確率 P_{il} で各発生分布が生じる場合に対する，各収集期間 j における偏りを表していることになる．

我々の手法では，以下の組をいくつか用意する．

(1) ホスト β_i
(2) 収集期間 J_j
(3) 特徴量 C_k
(4) 標本区間 S_l
(5) 発生確率 P_{il}

例えば，収集された HTTP の通信ログの通信を一ヶ月間収集し以下の組に対する標準偏差を求める．

(1) ホスト：通信元もしくは通信先ホストの IP アドレス毎
(2) 収集期間： 1 日毎

表 1　HTTP 通信における特徴量と標本区間の例

特徴量	標本区間
通信継続時間	継続時間の対数の切り捨て毎
リクエスト URI 長	URI 長の対数の切り捨て毎
Referer 長	リファラーヘッダ長の対数の切り捨て毎
ステータスコード	ステータスコード毎

(3) 特徴量：HTTP リクエスト URI の長さ
(4) 標本空間：長さの対数切り捨て
(5) 発生確率：一ヶ月分の平均値より計算

これまでの説明では，説明を単純化するためある通信データ X が，特定のホスト β_i，全収集期間を一定期間で区切った収集期間 J_j，唯一の標本区間 S_l に属し，特定の値（数値域）でのみ 1 になるかのように説明してきた．しかし，振り分け関数 f, g, h を工夫することでより複雑な指標を構成することも可能である．例えば，β_i をホスト名やドメイン名にする，収集期間 J_j を同じ曜日毎に振り分ける，などが考えられる．

次節に，特徴量とその標本区間の例について述べる．

3.2　特徴量と標本区間の例

本節では，HTTP 通信における幾つかの特徴量の例とその標本区間について述べる．HTTP 通信における特徴量と標本区間の例を，表 1 に示す．以下，それぞれ説明する．

3.2.1　通信継続時間

通信継続時間を特徴量とする場合，ある HTTP 通信が開始されてから接続が切れるまでの時間 T に対し，属する標本区間 S_l を決定する以下の振り分け関数 $f(T, l)$ を用いる．

$$f(T,l) = \begin{cases} 1\ (l\ =\ \lfloor \log T \rfloor) \\ 0\ (l\ \neq\ \lfloor \log T \rfloor) \end{cases}$$

この振り分け関数 f によって得られる標本区間は，リクエストされたリソースに対するサイズや種別に対する分布を反映すると考える．もし単純に HTTP 通信時間に対する平均と標準偏差を求めた場合，動画などサイズの大きなリソースへのアクセスに伴う長い継続時間が平均と標準偏差に大きな影響を与える．しかし，HTTP 通信時間全体の特徴を捉えるには，HTML ファイルに対する要求や，ステータスコードだけが返される場合など短い継続時間の通信の変動を考慮しなければならない．そこで，HTTP 通信継続時間の対数にて標本区間に振り分けることで，継続時間が長い通信だけでなく，短い通信についても考慮された指標を得ることができる．

3.2.2　リクエスト URI と Referer

リクエスト URI や Referer を特徴量とする場合，どのような文字が含まれるかは全く考慮せず，URL や Referer の長さ L だけに着目した以下の振り分け関数 $f(L, l)$ を用いる．

$$f(L,l) = \begin{cases} 1\ (l\ =\ \lfloor \log L \rfloor) \\ 0\ (l\ \neq\ \lfloor \log L \rfloor) \end{cases}$$

この振り分け関数 f によって得られる標本区間は，対象ホストにおける URI の分布やその木構造を反映すると考える．URI の長さだけに着目する理由は，効率性と直交性に優れると考えるからである．内容を考慮した特徴量は，計算に大量のリソースが必要とされ，また特徴量による評価が後段のセキュリティ異常検出器に影響を与えるかもしれない．我々はホワイトリストを作成するのが目的であり，完全なセキュリティ異常検出器を実現することではないため，より高速でかつ他の検出手法と独立性の高い指標を採用することにする．

3.2.3　ステータスコード

HTTP レスポンスに含まれるステータスコードを特徴量とする場合，ステータスコードの平均や標準偏差を求めても無意味と考え，ステータスコード毎に標本区間を構成し，その分布の標準偏差を求める．ある HTTP レスポンスにおけるステータスコードを H とすると，この振り分け関数 f は，以下の式で表される．

$$f(H,l) = \begin{cases} 1\ (l\ =\ H) \\ 0\ (l\ \neq\ H) \end{cases}$$

この振り分け関数にて得られる標本区間と発生分布において，ステータスコードの意味は全く考慮しない．例えば「200 OK」が返されるホストと，「404 Not Found」のみが返されるホストを比較した場合，ステータコードの意味に基づく場合「200 OK」が多く返されるホストが正常と一般的には判断される．しかし，我々の手法において常に同じステータスコードが返されるのであれば，それが例えサーバエラーを表す 500 であったとしても，定常的な通信であると評価される．

3.3　発生分布の標準偏差を考慮したホワイトリスト

我々の手法では，前節までに示した各特徴量における発生分布の標準偏差を使い，より通信数が多いホストを優先し構成した上で，定常性が低いホストを除外し，より定常的な通信を発生させるホストを追加する．ホワイトリストの作成手順を以下に示す．

(1) 一定期間の通信データを収集
(2) 収集された通信データからホスト毎の通信数を計算
(3) ホスト毎の各特徴量に基づく発生分布とその標準偏差を計算
(4) ホスト毎通信数の上位 N 位からホワイトリスト候補群を構成
(5) ホスト毎通信数の上位 $N + M$ 位からホワイトリスト予備群を構成
(6) ホワイトリスト候補群より全ての標準偏差が γ より大きいホストを除外

(7) ホワイトリスト予備群から全ての標準偏差が δ 以下のホストを候補群に追加

我々の手法では，定常的かつ頻繁にアクセスされるホストをホワイトリスト候補とし，発生分布の標準偏差にて評価した定常性の高いホストを補足する．標準偏差のみで評価した場合，例えば通信数が 1 しかないなど，通信数が少ないホストが優先的にホワイトリストに追加されてしまう．ホワイトリストは，より少ないホスト数にてより多くの通信を除去することが望ましいと考え，通信数が多いホストを優先的に含めることにする．

4. 実装と性能評価

本章では，提案手法に基づく実装と，2 つのデータを使った性能評価について述べる．

4.1 実装

本提案手法に基づき，ネットワークにて収集した HTTP 通信のデータより，各特徴量と標準偏差を計算するプログラムを実装した．その疑似コードをアルゴリズム 1 に示す．

この実装に適用したアルゴリズムは，3 章で示した標本分布の定義に従い全ての標本区間 S_l に対し愚直に振り分け関数を評価するのではなく，特徴量から直接的に属する標本区間を求めることで，ループの深さを一段減らし高速化を実現する．

今回の実装は，我々が所有する組織内ネットワークにおける HTTP 通信を解析したデータを読み込むことを前提とし，Perl 言語 (バージョン 5.10.1) にて実装した．次節では，この実装を用いた性能評価について述べる．

4.2 性能評価

実装したプログラムを用い，我々が収集した以下の 2 種の環境での HTTP 通信のデータを対象に評価を行う．

- 当機構の事務所ネットワークより収集した，組織内ネットワークデータ
- 当機構の運用する誘引環境 STARDUST より収集した，模擬環境ネットワークデータ

対象となる通信データは高野らの SF-TAP [8] による解析結果（JSON ファイル）である．解析結果ファイルの 1 行には，1 接続分の HTTP 通信に関する情報が含まれる．対象通信データのサイズを表 2 に示す．

このデータから標準偏差を求めホワイトリストを生成するまでの時間を計測した．性能評価に利用したマシンの仕様を表 3 に，処理時間の内訳を表 4 に示す．

この結果より，計算時間の大半が発生分布の計算であることがわかる．さらに，参考に記載したデータサイズ計測時の wc コマンドの結果との比較から，その実行時間の多くがファイルの読み込みに費やされる場合もあり，我々の実装が効率的な事が判る．また，一ヶ月分の通信データから 9 時間強で計算しており，実際の運用にて必要とされる周期，例えば一日毎のホワイトリスト更新も可能であり，十分な実用性を確保できると考える．

アルゴリズム 1 発生分布と標準偏差の計算

```
{ 発生分布を計算 }
for all すべての収集期間 j に対し do
  for all すべての通信データ X に対し do
    i ← X の通信先を求める
    a ← X の特徴量を求める
    l ← a の標本区間を求める
    R[i][j][l] を 1 増やす
  end for
end for
for all すべてのホスト i に対し do
  { ホスト毎に全期間でのデータ数を計算 }
  for all すべての収集期間 j に対し do
    for all すべての標本区間 l に対し do
      count_site_total[i] ← count_site_total[i] + R[i][j][l]
    end for
  end for
end for
for all すべてのホスト i に対し do
  for all すべての標本区間 l に対し do
    { 全標本区間に対し，全期間を平均した発生確率を計算 }
    for all すべての収集期間 j に対し do
      count_all_section_total[i][l] ←
        count_all_section_total[i][l] + R[i][j][l]
    end for
    P[i][l] ← count_all_section_total[i][l]/count_site_total[i]
  end for
  { 全ての発生分布に対する期待発生分布を計る }
  for all すべての収集期間 j に対し do
    for all すべての標本区間 l に対し do
      P_section_total[i][j] ← P_section_total[i][j] + R[i][j][l]
    end for
    for all すべての標本区間 l に対し do
      E[i][j][l] ← P_section_total[i][j] * P[i][l]
    end for
  end for
end for
for all すべてのホスト i に対し do
  δ ← 0
  num[i] ← 0
  { ホスト毎に全ての発生分布と期待発生分布の自乗和を計算 }
  for all すべての収集期間 j に対し do
    for all すべての標本区間 l に対し do
      δ ← δ + (E[i][j][k] − R[i][j][k])^2
      num[i] ← num[i] + 1
    end for
  end for
  { 標準偏差を計算する }
  D[i] ← √δ/num[i]
end for
```

5. ケーススタディ

5.1 ケース 1: インシデント対応支援向けホワイトリスト

マルウェア通信が発生しないことを前提とした組織内ネットワーク環境において，実際のセキュリティ機器が発

表 2　データサイズ

	期間	行数	バイト数
組織内ネットワーク	2019 年 7 月 1 日〜2019 年 7 月 31 日	1,551,513,916 行	459,874,440,176 byte
模擬ネットワーク	2018 年 8 月 1 日〜2019 年 7 月 31 日	15,405,934 行	152,894,294,354 byte

表 3　マシン性能

CPU	Xeon E5-2670 (2.6GHz)
コア数	8 コア
メモリ	96GB
OS	CentOS 6.10

表 4　処理時間の内訳

	組織内ネットワーク	模擬ネットワーク
発生分布の作成	537 分 53 秒	68 分 59 秒
期待発生分布の計算	8 分 38 秒	5 秒
標準偏差の計算	2 分 09 秒	1 秒
ホワイトリスト作成	1 分 52 秒	1 秒
合計	550 分 23 秒	69 分 06 秒
wc の実行結果 (参考)	139 分 17 秒	62 分 22 秒

表 5　アラート数とホワイトリストでの除去数

期間	アラート数	除去数	除去率
2019 年 4 月	31349	451	1.4%
2019 年 5 月	26161	785	3.0%
2019 年 6 月	9757	663	6.8%

生させるアラートに対し我々のホワイトリストを適用することで，実際のインシデント対応支援に即した評価を実施した．

5.1.1　収集データと収集期間

当機構にてインシデント支援のために運用されているセキュリティ機器として，株式会社 PFU の「iNetSec MP 2040」[9] を対象とし，2019 年 4 月〜6 月に検知したアラートを用いる．機構内ネットワークより 2019 年 3 月の一ヶ月間に収集された HTTP の通信データを用い，1000 の IP アドレスを含むホワイトリストを生成した．

5.1.2　生成ホワイトリストの評価

生成されたホワイトリストを，アラートに対し適用し除去可能であったアラートの件数を表 5 に示す．

発生したアラートは当機構のセキュリティオペレータが精査しており，これを踏まえ以下の結果が得られる．

- ホワイトリスト適用によって除外されたアラートは，すべてセキュリティ機器による誤検知であり，除去されても問題ないアラートであった．
- ホワイトリストを適用したとしても，致命的なアラートは除去されず，見逃を発生させない．
- ホワイトリストの適用は，アラートの削減に寄与する．

このように，我々のホワイトリストは，当機構における組織ネットワークに特化したホワイトリストとして有効に作用する．

5.2　ケース 2: 能動的観測支援向けホワイトリスト

マルウェア通信が発生することを前提としたネットワーク環境において，ホワイトリスト適用により定常通信を除去することで分析作業の効率向上を実現する，サイバー攻撃の能動的観測支援に即した評価を実施した．

5.2.1　収集データと収集期間

サイバー攻撃誘引基盤 STARDUST [10] には，誘引実験の際にネットワーク上の活動を観測するため，HTTP 通信など主要なプロトコルを解析し通信内容を表示する機能を Web UI 上に実装されている．この Web UI 機能に適用するホワイトリストの作成を試みた．

我々の手法が，マルウェア通信の観測に与える影響について評価するため，誘引実験期間を含むデータを収集データとした．2016 年 8 月に模擬ネットワークを利用した誘引実験を実施しており，2016 年 8 月の HTTP の通信データからホワイトリストを作成し評価する．

5.3　生成ホワイトリストの評価

得られたホワイトリスト候補の順位，データ数，対象通信データにホワイトリスト適用した際の除去率，標準偏差の計算結果を表 6 に示す．

計算結果に基づき，ホワイトリスト候補から全標準偏差が 0 より大きいホストを除外し (この表では 3 位と 4 位が候補からの除外対象)，HTTP 通信データの約 42.7%を除去可能なホワイトリストが作成された．得られたホワイトリストの上位には，定常的に発生している通信が現れている．この定常通信には，OS アップデートに伴う通信や CRL リスト取得など他環境においても発生しうる通信だけでなく，対象環境固有の通信も含まれる．例えば，1 位は環境内 Windows PC のウィルス対策ソフトが，環境内の Web API サーバに対し定期的に発生させる通信である．

ここで，7 位のホストは 8 月に観測対象となるマルウェアによる C&C サーバであり，その通信データは定期的なポーリングのみを含むため，標準偏差は 0 となり，ホワイトリストとして追加されることになる．我々の手法における標準偏差は，定常通信を判別する指標であり，その内容は問わないため，このような結果となった．

6.　考察

6.1　組織内ネットワークデータに対する考察

我々の手法における発生分布の標準偏差を分析するため，グラフを作成した（図 1）．

グラフの横軸はデータ数による順位であり，縦軸は順に

表 6　模擬環境ネットワークの通信データに対する計算結果 (一部のデータを加工)

順位	データ数	除去率	ステータスコード	Referer	URI	IP アドレス	備考
1	138421	39.0%	0.0	0.0	0.0	203.0.113.56	環境内の Web API サーバ
2	8485	2.4%	2185.9	0.0	371.4	160.26.2.187	jp.archive.ubuntu.com
3	2430	リスト除外対象	4062.5	1796.8	4341.3	203.0.113.36	環境内の Proxy サーバ
4	1747	リスト除外対象	0.2	0.3	0.4	203.0.113.44	環境内の Web サーバ
5	1623	0.5%	0.7	0.0	0.0	72.246.190.98	crl.microsoft.com
6	1489	0.4%	0.0	0.0	0.0	72.246.190.97	crl.microsoft.com
7	1485	0.4%	0.0	0.0	0.0	xxx.xxx.xxx.xxx	C&C サーバ
総データ数	354598	42.7%					(参考値)

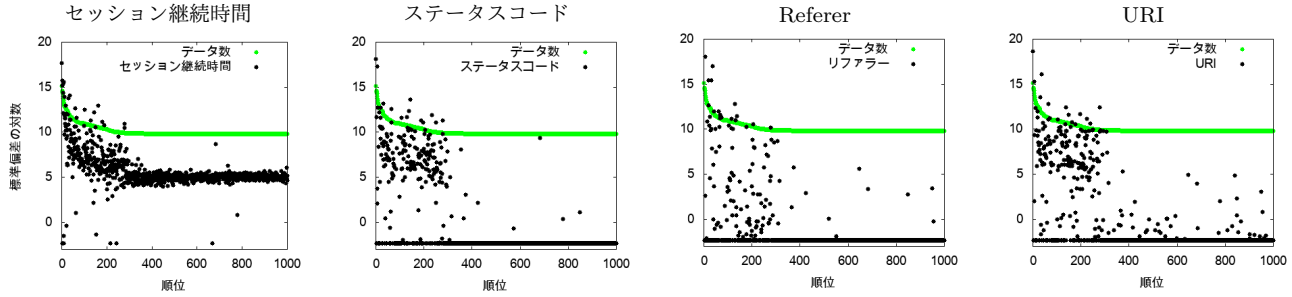

図 1　組織内ネットワーク通信データ標準偏差のグラフ

セッション継続時間，ステータスコード， Referer URI，リクエスト URI に対する標準偏差値の対数を表し，緑の点にてデータ数をプロットしている．

4 つのどのグラフにおいても約 400 位を境界として異なる分布となり，大半のホストにおいてステータスコード，Referer，URI の標準偏差が 0 となる．通信データを解析した結果，以下が要因である．

- これらの通信は，当機構が所有する NTP サービスに対する通信である．
- 当機構が所有する NTP サービスに対し，複数のホストから同じ HTTP リクエストが発生する．
- 複数のホストにおいてほぼ等間隔でリクエストが発生するため，累積リクエスト数がほぼ同じとなる．
- 同じ順位に同じリクエストを繰り返すホストが並ぶことで，標準偏差 0 となる直線が現れる．

これらのホストによる HTTP 通信のデータの例を図 2 に示す．これらのホストの HTTP 通信は，すべて時刻同期デーモンからの NTP サーバに対するポーリング通信から構成されている．これはまさに我々の目的としていた，定常通信を発見しホワイトリストとして排除する，という目的に合致した通信データ群を発生させるホストを発見することができた．この事実は，我々のホワイトリスト作成手法の実用性を実証している．

6.2　模擬ネットワークデータに対する考察

次に，模擬ネットワークでマルウェア通信を含む通信データから得られたホワイトリストについて考察する．

2016 年 7 月の通信データから生成されたホワイトリストは，模擬ネットワーク運用者の経験に基づき設定されたホワイトリストを包含しており，Web UI による表示にて不要な通信データ排除することで，着目すべき通信データに集中することができ，解析作業の効率向上に寄与すると考えられる．

```
{ "client":{"body":"","ip":"xxx.xxx.xxx.xxx",
  "time":1552165456.544498,
  "method":{"uri":"/","method":"HEAD"},
  "header":{"user-agent":"htpdate/1.1.1",
  "pragma":"no-cache","host":"ntp-a1.nict.go.jp",
   ...
{ "client":{"body":"","ip":"xxx.xxx.xxx.xxx",
  "time":1552165686.546661,
  "method":{"uri":"/","method":"HEAD"},
  "header":{"user-agent":"htpdate/1.1.1",
  "pragma":"no-cache","host":"ntp-a1.nict.go.jp",
  ...
{ "client":{"body":"","ip":"xxx.xxx.xxx.xxx",
  "time":1552165892.54833,
  "method":{"uri":"/","method":"HEAD"},
  "header":{"user-agent":"htpdate/1.1.1",
  "pragma":"no-cache","host":"ntp-a1.nict.go.jp",
   ...
{ "client":{"body":"","ip":"xxx.xxx.xxx.xxx",
  "time":1552165932.548977,
  ...
```

図 2　通信データの例（一部抜粋）

その一方で，2016 年 8 月の通信データから作られたホワイトリストは，マルウェアによる C&C サーバへのポーリング通信を定常通信として判別し，C&C サーバがホワイトリストに含まれてしまうことになる．この対策として，以下の 3 つの対策が考えられる．

(A) 実験開始前 1 ヶ月間のデータで作成したホワイトリストを適用する．

(B) 8 月に発見した C&C 通信については，9 月以降はブラックリスト化しホワイトリストから除外する

表 7　マルウェアのポーリング通信に対する標準偏差

特徴量	8 月のデータのみ	7 月の発生確率で計算
ステータスコード	0	276875
Referer 長	0	266875
リクエスト URI 長	0	27665

(C) 8 月のデータを使ったホワイトリスト作成時に，7 月以前の発生確率を使う．

このうち対策（C）を実装・評価した結果を表 7 に示す．この実装では 7 月のデータから発生確率を求め，8 月のデータにおける C&C サーバに対する標準偏差を計算した．

8 月の通信データのみで標準偏差を計算した場合と違い，7 月の発生確率で計算すれば 8 月に突発的に発生した C&C サーバとの通信が全て同じ特徴量であったとしても標準偏差値は 0 とならず，7 月の発生分布と比べ著しく偏っている非定常通信であると判断できる．

この発生確率を先月の通信データで計算する改良は，先月の通信データのみで計算する方式と比べ，ポーリング通信の突発的な変化に対応できるという利点がある．例えば，7 月の通信データで作成されたホワイトリストに定期的なポーリングの通信先が加えられていた場合，8 月に攻撃行動の変化等によりポーリングの内容が変化したとしても，ホワイトリスト適用の結果としてその変化を伴う通信が除外され，見逃すことになる．一方，7 月の発生確率を使い 8 月最新の通信データから作成したホワイトリストを適用すると，標準偏差が増加することでホワイトリストから除外され，突発的な変化の見逃を防ぐ事ができると考える．

7. おわりに

本稿では，対象となるネットワークの特性に特化したホワイトリストを作成する手法を提案した．本提案手法では，対象ネットワークから得られる通信データから複数の特徴量を求め，各ホスト毎に特徴量の発生分布とその標準偏差を計算する．発生分布の標準偏差を指標とすることで，大量の通信の特性を単純な数値として表現し比較可能にした．この標準偏差が小さいホストは定常通信のみを発生させるとし，標準偏差が小さいホストで構成されたホワイトリストを作成する．この手法に基づき，我々が所有する 2 つのネットワークから収集したトラフィック解析結果からホワイトリストを作成し，その有効性を評価した．その結果，我々の提案手法によるホワイトリストは，Alexa Top 500 などによる共通的なホワイトリストと比べ，対象組織固有の定常通信を反映したより実用的なホワイトリストであることを確認した．

今後の課題は，周期性の組み込みなど，より複雑な標本区間と振り分け関数の実現である．現状のアルゴリズムおよび実装では，Windows Update などの特定の日，曜日，時刻だけに発生する定期的な通信に対応できない．例えば，標本区間として 0 時から 23 時までの時刻を一時間毎に区切り，30 日分のデータに対する時刻による分布を求めこの標準偏差を計算することで，時刻に依存した定常通信を評価することができ，また標本区間毎の標準偏差からホワイトリストを時刻毎に変更することも考えられる．様々な特徴量に対する発生分布とその標準偏差の性質を解析し，ホワイトリストの応用範囲を広げたい．

謝辞　本研究の評価では，株式会社 PFU との共同研究として情報通信研究機構ネットワークに設置された iNetSec MP 2040 のアラートデータを利用した．ここに，株式会社 PFU 殿及び共同研究関係者の御協力に感謝の意を表する．

参考文献

[1] Theuns Verwoerd and Ray Hunt. Intrusion Detection Techniques and Approaches. *Computer Communications*, Vol. 25, No. 15, pp. 1356 – 1365, 2002.

[2] Keyword Research, Competitive Analysis, & Website Ranking — Alexa. `https://www.alexa.com/`.

[3] Seungoh Choi, Yeop Chang, Jeong-Han Yun, and Woonyon Kim. Traffic-Locality-Based Creation of Flow Whitelists for SCADA Networks. In Mason Rice and Sujeet Shenoi, editors, *Critical Infrastructure Protection IX*, pp. 87–102, Cham, 2015. Springer International Publishing.

[4] Younghwa Jang, Incheol Shin, Byung-Gil Min, Jungtaek Seo, and Myungkeun Yoon. Whitelisting for Critical IT-Based Infrastructure. *IEICE Transactions on Communications*, Vol. E96.B, pp. 1070–1074, 04 2013.

[5] Keisuke Takemori, Takahiro Sakai, Masakatsu Nishigaki, and Yutaka Miyake. Detection of Bot Infected PC Using Destination-based IP Address and Domain Name Whitelists. *Information and Media Technologies*, Vol. 6, No. 2, pp. 649–659, 2011.

[6] Ting-Fang Yen, Alina Oprea, Kaan Onarlioglu, Todd Leetham, William Robertson, Ari Juels, and Engin Kirda. Beehive: Large-scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks. In *Proceedings of the 29th Annual Computer Security Applications Conference*, ACSAC '13, pp. 199–208, New York, NY, USA, 2013. ACM.

[7] Tsong Song Hwang, Tsung-Ju Lee, and Yuh-Jye Lee. A Three-tier IDS via Data Mining Approach. In *Proceedings of the 3rd Annual ACM Workshop on Mining Network Data*, MineNet '07, pp. 1–6, New York, NY, USA, 2007. ACM.

[8] Yuuki Takano, Ryosuke Miura, Shingo Yasuda, Kunio Akashi, and Tomoya Inoue. SF-TAP: Scalable and Flexible Traffic Analysis Platform Running on Commodity Hardware. In *29th Large Installation System Administration Conference (LISA15)*, pp. 25–36, Washington, D.C., November 2015. USENIX Association.

[9] サイバー攻撃検知・SOC 運用効率化 iNetSec MP 2040. `https://www.pfu.fujitsu.com/inetsec/products/mp/`.

[10] 津田侑, 遠峰隆史, 金谷延幸, 牧田大佑, 丑丸逸人, 神宮真人, 高野祐輝, 安田真悟, 三浦良介, 太田悟史, 宮地利幸, 神薗雅紀, 衛藤将史, 井上大介, 中尾康二. サイバー攻撃誘引基盤 STARDUST. コンピュータセキュリティシンポジウム 2017 論文集, 第 2017 巻, oct 2017.