情報処理学会論文誌 Vol.62 No.12 1850–1863 (Dec. 2021)

複数組織の接続傾向を用いた自律進化型防御システムの提案と評価

西嶋 克哉[1,a)] 川口 信隆[1] 植木 優輝[1] 重本 倫宏[1] 近藤 賢郎[2] 中村 修[3]

受付日 2021年3月8日, 採録日 2021年9月9日

概要：サイバー攻撃の激化にともない，悪性サイトへの接続を防止する技術が求められている．既存手法としては，ブラックリストやホワイトリストを用いたものがある．しかし，公開されているブラックリストやホワイトリストだけで，インターネット上にあるすべての良性 Web サイト，悪性 Web サイトを網羅するのは難しい．したがって，これらリストに存在しないサイトが良性か悪性かを判断できないという課題がある．本稿ではこの課題を解決する手法を提案する．提案手法では組織が保有する，サイトへの接続ログを元に接続傾向を分析し，その傾向との乖離度を利用してサイトの良性，悪性を判定する．また，自組織の接続傾向だけでなく，他組織の接続傾向を利用することで判定の精度向上を狙う．さらに，判定で誤って悪性と判断した良性サイトへの接続を即時遮断するのではなく，機械には突破困難な追加認証を課し，突破できなかった場合のみ接続を遮断する．提案システムにより，業務遂行に必要なサイトを誤って悪性と判定した際の業務阻害を緩和しつつ，悪性サイトへの接続を遮断できることが期待できる．また，評価実験により，提案システムの有効性を評価する．

キーワード：悪性 Web サイト，追加認証，異常検知，情報共有

Proposal and Evaluation of Automated Defense System Using Access Tendency in Multiple Organizations

Katsuya Nishijima[1,a)] Nobutaka Kawaguchi[1] Yuki Ueki[1] Tomohiro Shigemoto[1] Takao Kondo[2] Osamu Nakamura[3]

Received: March 8, 2021, Accepted: September 9, 2021

Abstract: As cyber attacks intensify, technology to prevent connection to malicious sites is required. As an existing method, blacklists or whitelists are used. It is difficult to cover all benign websites and malicious websites on the internet with only public blacklists and whitelists. Therefore, there is a challenge that the websites which do not exist in these lists cannot be judged whether it is benign or malicious. In this paper, we propose the method to solve this challenge. In the proposed method, the access tendency is analyzed based on the access log to the site by the organization, and the site is judged to be benign and malicious by using the deviation from the tendency. In addition, it aims at improving the accuracy of the judgment by using not only the access tendency of the own organization but also the access tendency of other organizations. Furthermore, instead of immediately blocking the connection to a benign website determined to be malicious by the judgment, it imposes a difficult additional authentication for the machine and blocks the access only if it cannot be surpassed. By the proposed system, it is expected to be able to block the connection to a malicious website while mitigating the business inhibition when the websites necessary for the execution of business is wrongly judged to be malicious. In addition, the effectiveness of the proposed system is evaluated by the evaluation experiment.

Keywords: malicious website, additional authentication, anomaly detection, information sharing

1. はじめに

近年，標的型攻撃に見られるように，攻撃が高度化しており，企業や国家にとって重大な脅威となっている．ここで，マルウェアのダウンロード [1] に加えて，マルウェアとの通信，フィッシングサイトの表示，スパムの発信 [2] 等，悪意を持ったサイト（以降，悪性サイト）がサイバー攻撃において重要な役割を有している．このことから，サイバー攻撃の被害を抑制するためには，Firewall や Web Proxy 等のゲートウェイにより悪性サイトへの接続を遮断する，いわゆる出口対策 [3] が重要であるといえる．悪性サイトへの接続を遮断する方法としては，悪性サイトと判定したサイト群であるブラックリスト（以降，BL）への接続を遮断する方法や，反対に良性サイトと判定したサイト群であるホワイトリスト（以降，WL）以外への接続を遮断する方法が存在する．しかし，公開されている BL や WL だけで，インターネット上にあるすべての良性 Web サイト，悪性 Web サイトを網羅するのは難しい．したがって，これらリストに存在しないサイト群（以降，未知サイト）への接続が良性か悪性かを判定できないという課題がある．そこで本稿では，未知サイトへの接続の不審度を，接続傾向との乖離度から算出し，当該不審度の高低により未知サイトへの接続が良性か悪性かを判定する方法を提案する．これは，悪性サイトへの接続は，ふだんの接続傾向とは異なる可能性が高い，という考えに基づいている．なお，接続傾向は，組織が保有するサイトへの接続ログを分析し算出する．

上述の方法は，ふだんの接続傾向と異なる接続をすべて悪性サイトへの接続と見なす．このため特に，良性サイトを悪性サイトと判定してしまう誤検知が多く発生する．そこで誤検知の削減方法として，ふだんの接続傾向がより多くの良性サイトをカバーできるように，多種多様な正常の接続情報をインプットとして利用することが考えられる．ここで日立製作所では，慶應義塾大学と協力して，複数組織の SOC（Security Operation Center）をまたがったセキュリティ・オペレーション連携により，サイバー攻撃への集団防御を実現する，分散 SOC アーキテクチャを提案している [4]．本アーキテクチャの考えに基づき，自組織が保有する接続ログだけでなく，他組織が保有する接続ログを利用する．これにより，たとえば自他組織がともに不審と判断したときのみ悪性と判定する，といった方法で誤検知の削減が期待できる．これは，悪性サイトへの接続は，他の異業種組織であってもふだんの接続傾向と乖離している，という仮説に基づいている．一方で，組織が保有する接続ログは一般的に機微情報であり，かつログ量も大きくなる傾向があるため，他組織への共有は困難である．ここで我々は，生データの開示をともなわない分析を支援する，秘匿データ分析システムの研究を進めている [5]．本稿では前述のシステムを応用した，接続ログといった生データを直接共有するのではなく，他組織上で接続傾向の作成と接続傾向を用いた不審度の算出を行う分析ロジックを実行し，不審度のみを共有するシステムを提案する．

また，他組織の接続ログを用いることで精度が向上できたとしても，依然として誤検知の可能性は残っており，業務で利用する良性サイトが悪性サイトと判定され，接続が拒否されることにより，業務が阻害される可能性がある．そこで，接続先が悪性サイトと判定された際に，即座に接続を遮断するのではなく，いったん追加認証を課することにより，人間による業務上必要な接続は許可しつつ，マルウェア等による機械的な接続は遮断する．

これらにより提案システムは，未知の悪性サイトにも有効性があり，業務遂行に必要なサイトを誤って悪性と判定した際の業務阻害を緩和することが可能である．

本研究の主な貢献は以下の 3 点である．

- 機微性が高いセキュリティログを組織の垣根を越えて共有/分析するためのプラットフォームを設計/開発
- プラットフォーム上で動作し，複数組織の接続ログを基に，悪性ドメインを発見する異常検知アルゴリズムを考案．性能評価実験を通じ，自他組織両方の接続ログを使用することにより，検知性能を自組織の接続ログのみを使用した場合と比べて，Area Under the Curve（以降，AUC）の観点で 0.018 向上することを確認
- プラットフォーム活用のアプリケーションとして，著者らが研究を進めてきた悪性サイトへの接続防止手法である，自律進化型防御システム（AED：Autonomous Evolution of Defense）[6], [7], [8] および Web プロキシと連携して，Web 接続制御を行う機構を開発．ユーザ評価実験により制御処理の 93.6%が 3 秒以内であることから，実運用に適応可能であることを確認

本稿の構成は次のとおりである．まず，2 章で既存の悪性サイトへの接続防止手法とその課題について述べる．3 章で同課題を解決する手法を提案する．4 章で提案システムの実装，実環境を用いた評価実験および有効性の評価を行う．その後，5 章で関連研究について述べ，最後に 6 章でまとめを述べる．

2. 背景と課題

本章では，既存の悪性サイトへの接続防止手法である，

[1] 株式会社日立製作所研究開発グループ
Research & Development Group, Hitachi, Ltd., Yokohama, Kanagawa 244–0817, Japan
[2] 慶應義塾情報セキュリティインシデント対応チーム
Computer Security Incident Response Team, Keio University, Minato, Tokyo 108–8345, Japan
[3] 慶應義塾大学環境情報学部
Faculty of Environment and Information Studies, Keio University, Fujisawa, Kanagawa 252–0882, Japan
[a)] katsuya.nishijima.xn@hitachi.com

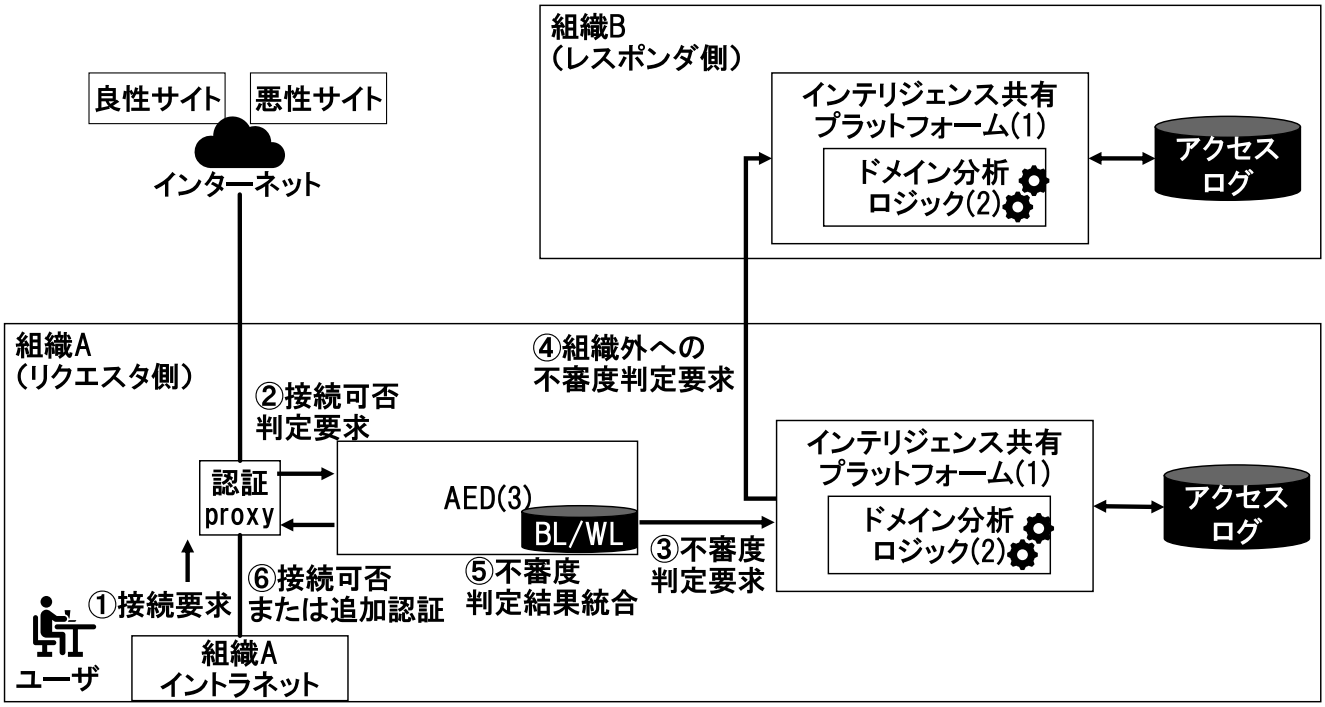


図 1 提案システム（分散異常検知型 AED）の全体像
Fig. 1 Overview of the proposed system (Decentralized outlier AED).

AED の概要とその課題を述べる．

2.1 AED の概要

AED では，ユーザが未知サイトへ接続しようとした場合に，プロキシで追加認証を要求する．これにより，マルウェアによる機械的な接続を遮断しつつ，人間による業務上必要な接続は許可することができる．また，WL 型 AED [8] では，組織に属する一定数のユーザが追加認証を突破し接続したサイトを WL に追加することにより，追加認証の発生回数を減らすことで，ユーザの利便性低下を抑えることができる．

2.2 AED の課題

(1) 未知サイト接続時の利便性低下

WL 型 AED では，WL を動的に拡張していく仕組みがあるが，未知サイトへ接続する際には必ず追加認証が発生してしまい，ユーザの利便性が低下する．

(2) 小規模組織で WL の拡張が困難

AED で WL を拡張する仕組みは，組織内の他ユーザが接続した記録を用いるため，ユーザ数が少ない小規模の組織では十分に WL の拡張が行えない．実際に先行研究 [8] では，ユーザが許容できる利便性を維持するために，1,000 人以上のユーザが必要という結果が出ている．これは大きな組織でないと実現が困難である．

3. 分散異常検知型 AED

2 章で述べた課題を解決するシステムとして，複数組織の接続傾向を利用して接続を制御する，分散異常検知型 AED を提案する．本章では，提案システムの設計について述べる．図 1 に提案システムの全体像を示す．本システムは，(1) インテリジェンス共有プラットフォーム，(2) ドメイン分析ロジック，(3) AED に分類される．

インテリジェンス共有プラットフォームは，データ分析を行う機能である分析ロジック（本ケースでは後述するドメイン分析ロジック）を他組織で実行することで，他組織が保有する生データである各種インテリジェンス（本ケースでは接続ログ）を直接取得せずに分析結果（本ケースでは接続の不審度）のみを取得可能とする．組織が保有する接続ログは一般的に機微情報であり，かつログ量も大きくなる傾向があるため，他組織への共有は困難である．前述の分析結果のみを取得することにより，効率的かつセキュアに他組織のデータを活用することが可能となる．このように，他組織の接続ログを使い，接続ログを記録するユーザ数を仮想的に増やすことで，2.2 節課題 (2) を解決する．なお，図ではレスポンダ側組織が 1 つの場合を例示したが，レスポンダ側組織が複数存在する構成も考えられる．

また，ドメイン分析ロジックは，異常検知アルゴリズムを利用し，組織が保有するサイトへの接続ログから接続傾向を分析し，あるサイトへの接続の不審度を，当該接続傾向との乖離度から算出する．これにより，WL によらない判定が可能となり，2.2 節課題 (1) を解決する．そして AED は，WL，BL，インテリジェンス共有プラットフォームから得られた不審度を元に，接続の可否や追加認証を制御する．

図 1 に示した，ユーザがインターネットへの接続を行う際の，分散異常検知型 AED の処理フローについて以下に説明する．

(1) 接続要求

組織 A のユーザが認証 Proxy を介してインターネットの Web サイトに接続要求を行う．

(2) 接続可否判定要求

認証 Proxy は，AED に対して，ユーザが要求した接続について，不審度判定を要求する．AED は接続先 Web サイトのドメインが WL や BL に含まれている場合は処理 (6) に移行する．WL，BL に含まれていない場合は処理 (3) に移行する．

(3) 不審度判定要求

AED は，組織 A 内のインテリジェンス共有プラットフォームに対して，接続の不審度判定を要求する．インテリジェンス共有プラットフォームは，組織 A のドメイン分析ロジックにより，組織 A の接続傾向を使用し不審度を算出する．

(4) 組織外への不審度判定要求

組織 A のインテリジェンス共有プラットフォームは，組織 B のインテリジェンス共有プラットフォームに接続先の不審度判定を要求する．インテリジェンス共有プラットフォームは，組織 B のドメイン分析ロジックにより，組織 B の接続傾向を使用し不審度を算出する．

(5) 不審度判定結果統合

AED は，組織 A，B が算出した不審度を元に，接続の不審度を判定する．具体的には，組織 A，B が算出した不審度のうち，値が小さい方があらかじめ定めた閾値を上回る場合，接続が不審であると判定する．

(6) 接続可否または追加認証

AED は認証 Proxy に対して，不審度判定結果である，接続の可否，または追加認証を返す．接続先サイトが WL に含まれる場合は接続を許可し，BL に含まれる場合は接続を拒否する．また，(5) の不審度判定の結果，接続が不審だと判定された場合は追加認証を要求し，そうでない場合は接続を許可する．

以下の節にて，分散異常検知型 AED の各コンポーネントの設計について詳述する．

3.1 インテリジェンス共有プラットフォーム

3.1.1 構成

図 2 にインテリジェンス共有プラットフォームの概要図を示す．組織 A には組織 B へ分析リクエストを送るリクエスタが在り，リクエスタには GUI のツールや AED 等からリクエスト情報が送られる．組織 B には分析リクエストを受信し処理を行うレスポンダ，および接続ログといった共有・分析対象のインテリジェンスが存在する．本プラットフォームでは，分析の実行主体である分析ロジックは，リ

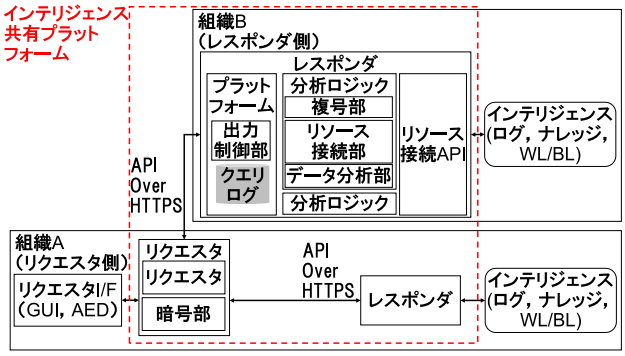


図 2 インテリジェンス共有プラットフォームの概要図
Fig. 2 Overview of the intelligence sharing platform.

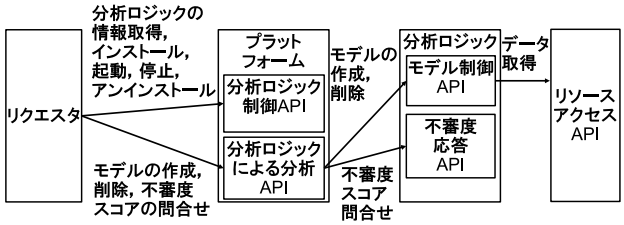


図 3 Web API の概要図
Fig. 3 Overview of the Web API.

クエスタ側が作成しレスポンダ側で実行される．レスポンダ内には，分析ロジックの実行基盤であるプラットフォーム，およびインテリジェンスへの接続制御を行うリソース接続 API が存在する．分析ロジックとその実行基盤を分離して別組織が管理することで，リクエスタ側の要望に応じて様々な分析を実施することができる．なお，レスポンダおよびインテリジェンスが他組織上に存在することを想定し設計をしているが，自組織内にも具備することで，他組織で行う分析と同様の分析を自組織の情報を用いて行うことも可能である．また，各コンポーネント間のやりとりは，WebAPI を介して行う．これについては，次節にて詳述する．また，本プラットフォームでは，リクエスタ側，レスポンダ側双方の情報流出を極力抑えるように設計している．セキュリティモデルの設計については，3.1.3 項にて詳述する．

3.1.2 WebAPI 設計

図 3 に WebAPI の概要図を示す．本節では，各 WebAPI の設計について述べる．

まず，プラットフォームの WebAPI について説明する．プラットフォームの WebAPI が提供する機能は，(1) 分析ロジックの制御，(2) 分析ロジックによる分析の実行，に分類される．

分析ロジックの制御では，分析ロジックの状態表示，インストール・起動，停止・アンインストール機能を提供する．また，リクエスタは分析ロジックによる分析の実行を，プラットフォームの WebAPI を介して実施する．これは，すべての接続をプラットフォームに集約させることによ

り，3.1.3 項にて詳述するセキュリティモデルにおいて効率性，網羅性を高めるためである．プラットフォームは，受け取ったリクエストからどの分析ロジックへのリクエストであるかを判断し，適切な分析ロジックへリクエストを転送する．これらの機能を持つプラットフォームは，各組織が共通のものを利用する．これにより，組織の本情報連携への新規参入を効率的にする．

次に分析ロジックが提供する WebAPI について説明する．分析ロジックが提供する WebAPI は分析ロジックごとに異なる．本稿で提案するドメイン分析ロジックでは，リクエスタ側は，レスポンダ側のインテリジェンス（接続ログ）を用いた分析モデルの作成・削除，分析モデルを利用したドメインの不審度判定を行うことができる．ここでいう分析モデルとは，不審度スコアを算出するのに使用するデータ構造であり，接続ログを前処理することで作成される．データ構造を事前に作成することで，分析リクエストに対して迅速に結果を返すことができる．作成されたデータ構造はレスポンダ内に保存される．この分析モデル作成のフェイズを以降「モデル作成フェイズ」と，分析モデルを用いて未知ドメインへの接続の不審度を算出するフェイズを以降「分析フェイズ」と呼ぶ．

最後に，リソース接続 API について説明する．リソース接続 API は，各分析ロジックがレスポンダ側の保有するインテリジェンスを取得する際に利用する WebAPI である．

3.1.3 セキュリティモデル

本節では，インテリジェンス共有プラットフォームのセキュリティモデルについて説明する．まず前提として “Trust-but-verify”，“Honest-but-curious” のモデルを採用する．Trust-but-verify というのは，相手のことを信頼するが，念のため検証を行うという考え方である．Honest-but-curious というのは，積極的に悪意を働くことはないが，容易な機会があれば不正を働く可能性があるという考え方である．インテリジェンス共有プラットフォームはインターネット上の不特定多数が参加するものでなく，身元が保証された組織間での利用を想定するため，前述の 2 つのモデルを前提としている．このため，リクエスタ側，レスポンダ側は互いが積極的に悪意ある行動を行う可能性は低いと判断しつつ，後に相手の行動を検証する手段を確保することを必要とする．

表 1 にリクエスタへの脅威を示す．まず，リクエスタへの脅威として，リクエストの内容が平文でレスポンダ側が取得可能なことがあげられる．そこで，リクエスタは分析ロジックとの間で共通鍵の交換を行い，暗号部によりリクエストを暗号化し，分析ロジックの複号部で復号してから分析を行う．これにより，後述するクエリログにリクエストが平文で記録されることを防ぐ．暗号に使う鍵は，DH [9] 等，Forward Secrecy を担保できるものを用いる．すなわち，暗号化されたトラフィックを受信していた攻撃者が鍵を入手した場合でも，過去のクエリを復元することをできないようにする．このリクエストの内容を保護する方法の注意点としては，レスポンダ側が管理している分析ロジック内に復号の鍵が存在するため，メモリの解析や分析ロジックのリバースエンジニアリングにより鍵がレスポンダ側に漏洩する可能性があることがあげられる．したがって，より強固にリクエストを保護したい場合は，組織間のデータを開示することなく，共通する要素だけを得るプロトコルである Private Set Intersection のような，リクエスタ側で暗号化したリクエストを複号せずに分析を行う方法を選択すべきである．ただし，暗号化した状態での分析には，分析手法に制限があることや，分析処理時間が増加するといった欠点があるため，実施したい分析とリクエストの秘匿化必要性に応じて，手法を選択すべきである．

表 1 リクエスタへの脅威
Table 1 Threats against requester.

#	分類	脅威
1	機密性	リクエスト内容の漏洩
2	完全性	レスポンダによる分析ロジックの改竄
3	完全性	他の分析ロジックが不正にリクエスタの分析ロジックに接続する

表 2 レスポンダへの脅威
Table 2 Threats against responder.

#	分類	脅威
1	機密性	分析ロジックによる必要以上のインテリジェンス取得
2	機密性	分析ロジックによるレスポンダのデータ漏洩
3	機密性	第 3 者によるシステムの利用
4	完全性	分析ロジックによるレスポンダのシステムへの不正接続
5	可用性	分析ロジックによるリソース（CPU, メモリ，DB 接続）大量消費

次に，レスポンダ側により分析ロジックが改ざんされるリスクがある．本リスクに対しては，分析ロジックを難読化することで，改ざんを防ぐことができる．また，分析ロジックに電子署名を付与し，それを検証することにより，分析ロジックの改ざんやレスポンスの改ざんを防ぐことができる．

最後に，分析ロジックは他の分析ロジックやレスポンダに不正アクセスされるリスクがある．本リスクに対しては，分析ロジックへの接続をプラットフォームからのみに制限する．また，分析ロジックに対して不正アクセスを検知する仕組みや，脆弱性管理，不正アクセスされた際にイメージから分析ロジックを復元する機能を導入することが対策として考えられる．

表 2 にレスポンダへの脅威を示す．まず，レスポンダへの脅威として，分析ロジックが分析に必要とする以上の情報を取得するリスクがある．本リスクに対しては，リソース接続 API で分析ロジックが接続可能なインテリジェンス

を制御することで対策する．また，分析ロジックに付与された電子署名を検証することにより，分析ロジックを認証し，認証の結果に応じて分析ロジックの認可を制御する．また，許可されていないリソース接続 API への接続をリアルタイムに監視することで，異常を検知することができる．その他，リソース接続 API への接続ログを保管することで，監査することができる．

次に，分析ロジックがレスポンダから取得したデータを漏洩させるリスクについて説明する．本リスクに対して，分析ロジックと外部との通信は必ずプラットフォームを介するようにすることで，一元的な監査・制御を行う．具体的には，リクエスタのリクエスト，分析ロジックのレスポンスをクエリログとして保管しておき，必要に応じてリクエスタ側にリクエストの復号・開示を要求できるようにする．なお，リクエスタ側は，開示要求に応じるために，暗号化に利用した鍵を保管しておく必要がある．また，出力制限部によって，分析ロジックから外部へ送られるデータ量を制限する．たとえば次節で説明するドメイン分析ロジックでは，レスポンダが送信する最低限のデータは不審度判定結果の数値のみであり，接続ログのデータ量と比較して非常に小さい．したがって，送信可能なデータ量を最低限必要なデータ量に制御することで，接続ログデータの漏洩を抑えることが可能となる．ただし，本機能の制限として，少量のデータを多数漏洩させる方法が考えられる．しかし，レスポンスログの数や内容を確認することで，不正な挙動を検知できる．

また，インテリジェンス共有プラットフォームが，本プラットフォームに参加をしていない不特定多数に利用されるリスクについて説明する．本リスクに対しては，プラットフォームの WebAPI に認証機能を具備することで対策する．

次に，分析ロジックがレスポンダのシステムに不正アクセスするリスクについて説明する．本リスクに対しては，レスポンダのシステムで不正侵入を検知する仕組みや，脆弱性管理，不正アクセスされた際にイメージから復元することで対策する．

最後に分析ロジックがリソースを大量に消費し，レスポンダのサービスを妨害するリスクについて説明する．本リスクに対しては，分析ロジックが利用可能な CPU・メモリ量を制限することにより，分析ロジックのリソース消費を制御する．また，分析ロジックがリソース接続 API を利用できる回数，同時接続数を制限することで，インテリジェンスを格納しているデータベースのリソース消費量を制御する．

3.2 ドメイン分析ロジック

本節で述べるドメイン分析ロジックで用いる異常検知アルゴリズムは，著者らが研究してきた AED [8] をベースに，文献 [10] にある N-gram による分析を追加で援用している．

AED 同様，本ロジックでは組織内で記録された Web 接続ログを基に当該組織における正常接続モデルを構築する．そして，入力として与えられた分析対象ドメインと当該モデルとの「距離」を基に異常検知を行う．

モデル化に際してはセキュリティ専門サイト等が公開している既知の悪性ドメインの情報は使用しない．これは，本研究の目的が，各々の組織で観測・記録するセキュリティ情報を，他組織と共有することによる検知効果を，組織単独のときと比較して評価するためである．

3.2.1 分析ロジックの要件

自組織だけではなく他組織上で実行されるという特徴から，分析ロジックには (1) 使用するメモリを一定量以内に抑える，(2) CPU 負荷を一定内に抑える，(3) インターネットへの通信を制限する，(4) 必要に応じてレスポンダが分析内容を監査できるようにする，(5) 必要な情報の一部がレスポンダ上にない場合でも，分析を実施できるようにする，という固有要件がある．後述の設計では，上記要件をどのように満たしているのか言及する．なお，要件 (4) については，3.1.3 項にて言及済みである．

3.2.2 入力

本実装では接続ログの各エントリは**表 3** のカラムを持つことを想定する．

表 3 接続ログのカラム
Table 3 Access log columns.

#	カラム	例
1	接続日時	2020-02-01T12:00:00:00.000
2	接続元の識別子	192.168.0.2
3	接続先サイトのドメイン名	www.example.com
4	ドメイン名に対応する IP アドレス	10.0.0.10

接続日時，接続元の識別子，接続先サイトのドメイン名，ドメイン名に対応する IP アドレスはいずれも一般的な Web Proxy ログ，DNS ログから取得できる情報である．分析フェイズにおいてリクエスタから送信される分析リクエストに含まれるクエリは，リクエスタ側で接続が発生した日時，問合せ対象ドメイン名，ドメイン名に対応する IP アドレスの 3 つであり，接続元識別子は必要としない．

3.2.3 ドメインの良性・悪性判定

図 4 にドメインの異常検知アルゴリズムの概要を示す．アルゴリズムは，接続に関する情報（接続日時，ドメイン，IP アドレス）を入力として与えられ，不審度を出力する．ドメイン悪性判定技術の中には WHOIS 情報やレジストリ情報を問い合わせるためにインターネットへの接続が必要なものも多いが [11], [12]，本アルゴリズムは追加情報を取得するための外部接続を必要としない．これにより 3.2.1 項の要件 (3) が満たされる．本アルゴリズムは「頻度ベー

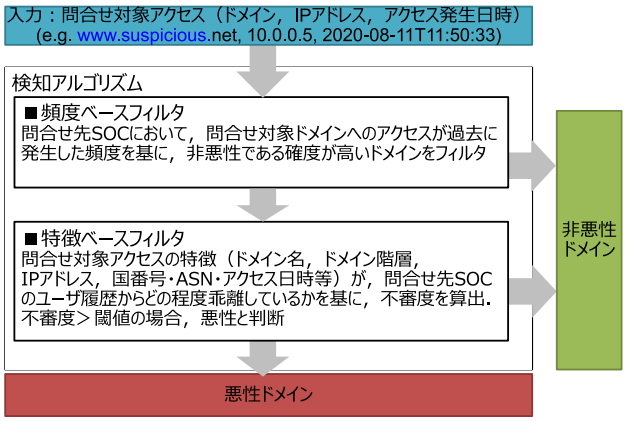


図 4 異常検知アルゴリズムの概要
Fig. 4 Overview of the outlier algorithm.

スフィルタ」，「特徴ベースフィルタ」の 2 段階から構成される．

頻度ベースフィルタは，リクエスタ側組織において，問合せ対象ドメインへの接続が過去に発生した頻度を基に，非悪性である確度が高いドメインをフィルタリングする処理である．たとえば，問合せ対象ドメインに対して，当該組織内の多数のユーザから過去何度も接続があるならば，当該ドメインは良性である確度が高いと判断できる．同様に，悪性ドメインの生存期間は一般的に短いことから，古くから接続していた履歴があるドメインは良性である確度が高いと判断することができる．問合せ内容がこのフィルタに合致した場合，アルゴリズムは不審度スコア = 0.0 として良性判定を返して処理を終了する．一方，問合せ内容がフィルタに合致しない場合は，次の特徴ベースフィルタに処理が移る．

特徴ベースフィルタは，問合せ対象接続の特徴（ドメイン名，ドメイン階層，IP アドレス，国番号・ASN，接続日時等）が，問合せ先組織での接続履歴と比べてどの程度乖離しているのかを基に，不審度スコアを算出する．不審度スコア算出は，(1) 近傍性，(2) カテゴリ類似性，(3) ドメイン名類似度という 3 つの観点を基に行う．(1) 近傍性では，(a) 問合せ対象ドメインと 3rd level ドメインが一致する別ドメインに接続したことがある自組織内ユーザ数，(b) 問合せ対象 IP アドレスと /24 で一致するアドレスに接続したことがある自組織内ユーザ数，を基に近傍性を算出する．(2) カテゴリ類似性の観点では，過去接続の各特徴（接続した時刻や接続ドメインのトップレベルドメイン等）をカテゴリとして分け，問合せ対象接続がどのカテゴリに属するかにより，接続の類似性を算出する．(3) ドメイン名類似度の観点では，ドメイン名の n-gram を基に，自組織が過去に接続したことがあるドメイン名との類似性を算出する．

3.2.4 疑似コード

次に，リスト 1 に示す疑似コードを用いてアルゴリズムの詳細について説明する．アルゴリズムの入力には，先述のドメイン（q_domain），IP アドレス（q_IP），接続日時（q_time）以外に，Country Code（q_CC），Autonomous System Number（q_ASN）がある．q_CC，q_ASN は q_IP を基に解決することができる．具体的なライブラリ名は後述する．それ以外にも，アルゴリズムで用いる閾値として，TH_hosts，TH_day，TH_close，w_closeness，w_fitness，w_normality がある．w_closeness，w_fitness，w_normality は特徴ベースフィルタの各観点の重みづけを示すものであり合計値は 1.0 となる．Dataset はモデル作成に使用する，レスポンダ側組織での接続ログを指す．また N は後述の n-gram で用いる値である．

1 行目～3 行目は頻度ベースフィルタの処理を示す．2 行目にあるように，問合せ対象の q_domain に接続したことがある Dataset 内ユーザ数が TH_hosts を超えるか，q_domain への接続履歴が TH_day より前にある場合，q_domain は良性である確度が高いと判断し，不審度スコア = 0.0 を返す（3 行目）．これは，多くのユーザが接続するドメインまたは古くから接続があるドメインは良性である確度が高いという知見からくるものであり，既存研究 [11] や我々の過去の研究 [8] でも活用されている．分析フェイズで q_domain と Dataset 内ユーザの突合を高速に行うために，分析フェイズであらかじめドメインごとの接続元ユーザ数や接続日時を記録する必要がある．このとき，問合せ先組織のユーザ数や接続ドメイン数によっては多くの記憶領域を要する必要がある．そこで我々は，記憶領域のメモリサイズを一定に保つために，ドメインごとの接続ユーザ数や TH_days より過去に接続があるドメインの一覧を，Count-min Sketch [13] および Bloom Filter [14] を用いて管理することにした．Count-min Sketch は key に対応するカウント数を確率的に保持する機構であり，key 数が増えてもメモリサイズは変わらない．同様に Bloom Filter はアイテム一覧を一定のメモリサイズで保持する機構である．これにより，分析ロジックの使用メモリサイズを一定以内に抑えるという 3.2.1 項の要件 (1) を解決する．

また，Count-min Sketch，Bloom Filter ともに処理に必要な計算量は key 数やアイテム数によらず一定であり，要件 (2) も満たす．なお，両者の課題として，保持する key 数あるいはアイテム数が増えるほど，実際には保持していない key やアイテムを保持していると誤判断する false positive の発生頻度が高くなることがあげられる．

4 行目～7 行目は，特徴ベースフィルタの中の近傍性に関する処理である．ここでは，q_domain の 3rd level ドメインおよび q_IP の /24 アドレスへ接続したユーザ数をカウントし，その数に応じて closeness を計算する．ユーザ数 ≧ TH_close のとき closeness = 1.0 となる．この処理は，一定数以上のユーザから接続履歴があるドメインと同じ階層下にあるドメイン，あるいは接続履歴がある IP アドレス

```
1   #q_domain にアクセスしたユーザ数が TH_hosts を超える，
    もしくは TH_day より過去にアクセスがある場合，無条件に score = 0 とする
2   if | {h | h in Dataset and h.hasAccesed(q_domain)}| > TH_hosts
    or | {h | h in Dataset and h.hasAccesedBefore(q_domain,TH_day)}| > 0:
3           return score = 0.0
4   #q_domain の 3rdlevel domain または g_IP/24 にアクセスしたユーザ数に応じて，
    通常アクセス先との近さ closeness を計算.
5   domain3rd = get3rdLevelDomain(q_domain)
6   Closeness = | {h | h in Dataset
    and h.hasAccesed(domain3rd)}| + |{h | h in Dataset and h.hasAccessed(g_IP/24)}|
7   Closeness = min(Closeness/TH_close, 1.0)
8   #CC, AN, TLD, 日中/夜間, weekday/weekend, ドメイン階層数・サブドメイン長から，
    カテゴリ観点での fitness（正常度合）を計算
9   domain1st = getTopLevelDomain(q_domain)
10  type_of_hour = getTypeOfHour(q_time)
11  type_of_day = getTypeOfDay(q_time)
12  domain_length_range = log16(maximum length of subdomaind of g_domain)
13  domain_hierarchy_range = log2(#of dot in g_domain)
14  fitness = WDOD(q_CC, q_AN, domain1st, type_of_hour,
    type_of_day, domain_length_range, domain_hierarchy_range)
15  #2ndlevel domain 以降の各レベルのドメインを対象とした n-gram の rank の観点，および階層数
    から，ドメイン名に関するデータセットとの類似度を計算する
16  for token in getN-gram(q_domain):
17          name_rank += log2(rank(token)) / log2(#unique tokens in Dataset)
18  normality = 1.0 - name_rank / (#tokens in g_domain)
19  #ネットワーク上の近さ，およびカテゴリ観点の異常度を基に最終的な outlier score を計算する
20  return
    score = 1.0-(w_closeness*closeness+w_fitness*fitness+w_normality*normality)
```

リスト 1 異常検知アルゴリズムの疑似コード
List 1 Pseudocode of the outlier algorithm.

と同サブネット下の IP アドレスは良性である可能性が高いというヒューリスティックに基づいている．モデル作成フェイズでは，Count-min Sketch を使用して 3rd level でのドメイン接続ユーザ数および/24 での IP アドレス接続ユーザ数をカウントしておく．

8 行目～14 行目は，特徴ベースフィルタの中のカテゴリ類似性に関する処理である．ここでは，IP アドレスの国名・AS 番号，ドメインの TLD・階層数・サブドメインの最大ラベル長，接続日時の日中/夜間・平日/週末，という 7 種類の特徴量を基に，Dataset 内の履歴と比較した，q_IP，q_domain，q_time の類似度 fitness を求める．ここで，各特徴量は numerical ではなく categorical なデータとして処理する．たとえば，日中/夜間の違いや AS 番号の数値に大小関係はなく，カテゴリとして処理するのが好ましい．階層数，ラベル長は数値として扱うことも可能だが，今回は階層数が浅い（2 以下）/深い（3 以上），ラベル長が一定値（16）以上/未満，という区分にわけ，カテゴリとして扱うことにする．また，GeoIP [15] を使うことでインターネット接続なしに，IP アドレスから国名・AS 番号を求めることができる．

モデル作成には one-class SVM のような機械学習を用いる方法も考えられるが，演算コストを抑えるために本研究

では Weighted density-based outlier detection（WDOD）を用いる [16]．WDOD は，特徴量ごとに各カテゴリの相対発生頻度を求めることで入力データの類似度を算出する，特徴量間の依存関係を考慮しない，という特性がある．特徴量間の依存関係を考慮しないというのは精度上の欠点になりうるが，本研究のように，他組織上で動作する分析ロジックでは 2 つの理由でそのシンプルさが功を奏する．1 つは WDOD の計算は軽量であることがあり，もう 1 つは Dataset が複数のログに分かれていた場合でも演算を行えることである．たとえば，問合せ先組織で使用できるデータセットが〈ユーザ ID, 接続先ドメイン, 接続日時〉,〈ユーザ ID，接続先 IP アドレス，接続日時〉の 2 つログに分かれている場合を想定する．この場合ドメイン名と IP アドレスを 1 対 1 に紐づけることはできず，モデル作成フェイズで名前解決をするのは 3.2.1 項の要件 (2)，(3) に反することになる．しかし WDOD では特徴量ごとに演算を行うため，ドメインと IP アドレスが紐づいていなくとも類似度の算出が可能である．以上，分析に必要な特徴量間の紐づけが不要となることで，3.2.1 項の要件 (2)，(3) および (5) を満たす処理が可能になる．理論上，ドメイン名，IP アドレス，接続日時は別々のログとして与えられたとしても処理を実行することが可能である．モデル作成フェイズでは WDOD のモデル作成を行い，分析フェイズではクエリと WDOD モデルとの類似度を求める．

15 行目～18 行目はドメイン名類似度 normality の算出処理であり，Dataset 内にあるドメインと q_domain との類似度を求める．ドメイン名類似度は，q_domain の各ラベルの n-gram が，Dataset 内のドメインの n-gram に含まれる頻度を基に算出する．ここで n-gram とは，ある文字列を n 文字ごとに区切ったサブ文字列の集合である．たとえば，“hitachi” の 3-gram は “hit”，“ita”，“tac”，“ach”，“chi” の 5 つである．ドメイン名の類似度に着目した既存研究としては文献 [10], [17] 等がある．特に文献 [10] は本研究と同じく n-gram を用いている．違いとしては，本研究では類似度の値を 0～1 の間に入るよう正規化するために出現頻度のランクを求めている点がある．モデル作成フェイズでは Dataset 内のドメインの n-gram の抽出およびランク付けを行い，分析フェイズでは q_domain の n-gram のランクを求める．

19 行目～20 行目は，前段で求めた closeness，fitness，normality を基に不審度スコアを算出する．算出にあたっては各値を w_closeness，w_fitness，w_normality で重みづけし加重平均を求める．前述のとおり不審度スコアは 0～1 の値をとり，値が大きいほど不審度が高いことを指す．

3.3 AED

AED は未知サイトへの接続が発生した際に，自他組織内のドメイン分析ロジックに不審度の問合せを行う．得られた不審度のうち，最小のものがあらかじめ定めた閾値を超えた際に，悪性サイトへの接続であると判定し，追加認証を行う．これにより，自組織で不審と判定された場合でも，他組織の観点で正常であれば正常と見なされ，誤検知を削減できる．

なお，AED では追加認証として人間と機械を判別するチューリングテストである Completely Automated Public Turing test to tell Com-puters and Humans Apart（以降，CAPTCHA）認証を課する．CAPTCHA により，人間による業務上必要な接続は許可しつつ，マルウェア等による機械的な接続は遮断する．CAPTCHA 認証には複数の種類がある．そのなかで，ユーザが回答することが容易で，広く普及している，画面に表示された文字を入力する方式を採用している．一方で，本方式の CAPTCHA 認証を突破する研究 [18], [19] も行われている．AED では追加認証の方式は可換であるため，ユーザの利便性低下とのバランスを考慮しながら，より突破困難な方式を採用することも可能である．

4. 評価

4.1 実装

図 2 記載のリクエスタ，プラットフォーム，分析ロジックは，それぞれ Python の Web アプリケーションフレームワーである Flask [20] を用いて作成し，Docker [21] のコンテナで実行した．

表 4 に記載した認証 Proxy および AED はあわせて 1 台のサーバ上に実装した．また，インテリジェンス共有プラットフォームも 1 台のサーバ上に実装した．また，各サーバは仮想マシンで作成した．表 4 に各サーバの性能を示す．

4.2 データセット

組織 A，組織 B が持つ接続ログを用いて，それぞれの組織の接続傾向である分析モデルを作成した．組織 A は AED を利用する 10 人程度の小規模組織で，組織 B は 1,000 人以上の大規模組織のデータを利用した．モデル作成に利用する接続ログには，dns ログと traffic ログがあり，dns ログには，接続日時，接続元の識別子，接続先サイトのドメ

表 4 各サーバの性能
Table 4 Spec. of each server.

サーバ用途	CPU コア数	Memory [GB]
認証 Proxy + AED	4	4
インテリジェンス共有プラットフォーム（リクエスタ側）	8	16
インテリジェンス共有プラットフォーム（レスポンダ側）	2	8

表 5 モデル作成に利用した接続ログ
Table 5 Access logs used to create the model.

	取得期間	件数/サイズ（dns ログ）	件数/サイズ（traffic ログ）
組織 A	2021 年 1 月 26 日～2021 年 2 月 5 日（11 日間）	249,712 / 約 20MB	247,735 / 約 50MB
組織 B	2021 年 1 月 26 日～2021 年 2 月 5 日（11 日間）	31,132,557 / 約 3GB	83,136,528 / 約 18GB

表 6 テストデータ
Table 6 Test data.

	取得期間	件数	情報源
良性サイト群	2021 年 2 月 8 日～2021 年 2 月 12 日（5 日間）	1,930	組織 A が接続したドメイン
悪性サイト群	2021 年 2 月 8 日～2021 年 2 月 12 日（5 日間）	1,620	PhishTank から収集したドメイン

イン名が，traffic ログには接続日時，接続元の識別子，ドメイン名に対応する IP アドレスが記録されている．**表 5** に各ログの取得期間，件数およびサイズについて示す．なお，組織 B の dns ログ，traffic ログは，実験環境の都合上送信元の識別子が固定されていない．ドメイン分析ロジックでは，送信元の識別子と接続先サイトの組合せが同じ接続ログは，1 つしかモデル学習に用いない．一方送信元識別子が変わると，別の接続ログとして認識され再度モデル学習に用いられる．このため送信元識別子が固定化されている場合と比較すると結果に違いが生じるが，接続傾向を利用した不審度判定という目的の実現には大きな影響がないと考えられる．

推定器は組織 A の接続が良性か悪性かを判定するのに使われるため，良性サイトのテストデータは，モデル化に使用したデータとは異なる期間に，組織 A が接続したドメインを用いた．使用した良性データのなかに悪性データが含まれないことは，確認済みである．また，悪性サイトのテストデータは，PhishTank [22] から収集したサイト群を用いた．悪性データの取得期間については良性データと合わせた．ドメインに悪性データの特徴を有していることが要件であり，仮に別期間（モデル作成に用いた期間等）であったとしても，評価結果の一般性は失われない．なお，本推定器の性能は対象ドメインの生存期間に依存しないため，悪性データセット作成に際してもこの点は考慮しない．それぞれのデータについて**表 6** に示す．

4.3 評価項目

評価項目は以下のとおりである．評価は従来技術である WL を用いた AED（以降，WL 型 AED），WL 型 AED の機能に加え，WL に含まれない接続に対して，単一組織の分析モデルにより異常検知を行う AED（以降，異常検知型 AED），および組織 A，B の分析モデルを用いる異常検知型 AED（以降，分散異常検知型 AED）の 3 パターンについて実施した．また，以降それぞれの方式の英語表記を，WL，Outlier，DeOutlier とする．

4.3.1 悪性サイトの検知精度

表 6 に示したテストデータを用い，異常検知型 AED および分散異常検知型 AED の悪性サイト検知精度について評価する．

4.3.2 ユーザの利便性

AED を組織 A で 2021 年 2 月 8 日～2021 年 2 月 26 日の 19 日間運用した結果得られたログを元に，CAPTCHA 発生回数を算出し，ユーザの利便性を評価する．なお，AED では N 人以上が CAPTCHA を突破して接続したサイトは WL に追加され，次回接続時には CAPTCHA が発生されない．本実験では，$N = 1$ とした．これは，実験を実施した組織 A の人数が少ないためである．その他，AED は HTTP リクエストに referer ヘッダがある場合は CAPTCHA を発生させない．したがって，接続したドメイン数に対して，CAPTCHA 発生回数は少なくなる．

また，異常検知を用いる場合，設定する閾値によって CAPTCHA の発生頻度が異なる．本実験では，全悪性サイトのうち，正しく悪性サイトとして検知した割合である検知率が 99，95，90%以上となるように閾値を変更して比較を行った．なお，テストデータの良性ドメインと悪性ドメインに重複はないため，WL 型 AED の検知率は 100%である．

4.3.3 処理性能

提案手法はフォワードプロキシと連携しており，ユーザが Web 接続を行うたびに不審度スコアを算出し，不審度が一定値以上の場合に追加認証を発生させる．この処理により Web 接続の処理時間が増えるため，その値が実用範囲内かを評価する．

評価は一番処理時間がかかる分散異常検知型 AED に対して行う．分散異常検知型 AED は処理速度短縮のため，以下の機能を実装している．

(1) キャッシュ機能

不審度判定に用いる入力項目である，接続が発生した日時と問合せ対象ドメイン名の組合せが過去の問合せと同一の場合，キャッシュしたスコアを返す．

(2) 組織外への問合せ抑制機能

組織 A の分析モデルを用いて不審度判定を行った結果，スコアが 0 の場合組織 B の分析モデルへの問合せを行わない．これは，分散異常検知型 AED では，組織 A，組織 B 双方のスコアのうち値が小さい方を採用するため，スコアの最低値である 0 が取得された時点で組織 B への問合せは意味を成さないためである．

4.4 評価結果

評価結果を示す．Receiver Operating Characteristic（以

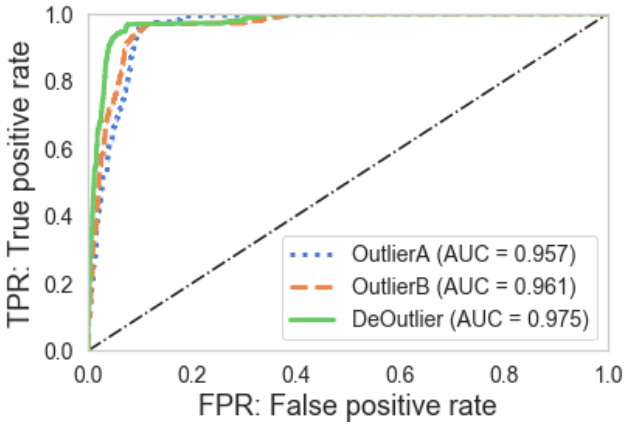


図 5 各方式の ROC 曲線
Fig. 5 ROC curve of each method.

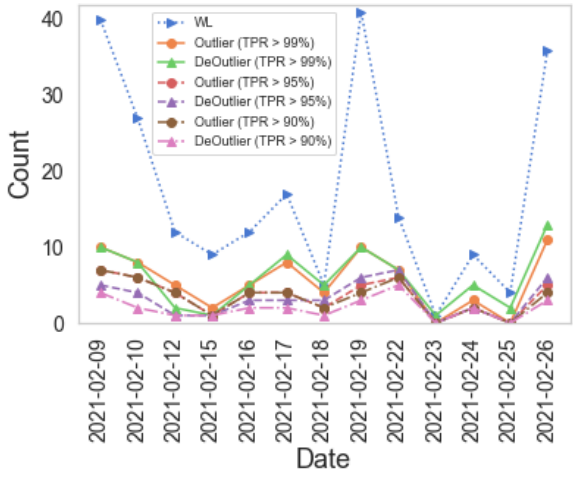


図 6 1 日ごとの CAPTCHA 発生数
Fig. 6 The number of daily CAPTCHA occurrences.

降，ROC）の算出には Scikit-learn [23] を，グラフ作成には Matplotlib [24] を用いた．

4.4.1 悪性サイトの検知精度

図 5 に表 6 に記載したデータセットを利用し算出した，異常検知型 AED と分散異常検知型 AED の ROC 曲線を示す．なお，異常検知型 AED は，組織 A の接続ログを用いた場合（OutlierA）と組織 B の接続ログを用いた場合（OutlierB）を評価した．ROC 曲線下の面積を AUC と呼び，AUC が 1 に近いほど，識別精度が高いことを示す．異常検知型 AED の場合 AUC は，組織 A の接続ログを使用した場合 0.957，組織 B の接続ログを使用した場合 0.961 であり，分散異常検知型 AED の場合 AUC は 0.975 であった．このことより，提案手法である分散異常検知型 AED が，他と比較し高い精度で悪性 Web サイトの判定ができることが分かる．

また，TPR（検知率）が 0.97 付近以上の箇所では，組織 A の接続ログを使用した異常検知型 AED の FPR（誤検知率）が他と比較し低くなることを示している．

4.4.2 ユーザの利便性

図 6 に土日祝日を除いた日ごとの CAPTCHA 発生数を示す．また，運用初日は WL 型 AED で大量の誤検知が発生するため，これも除外した．また，**表 7** に CAPTCHA 発生数の統計データを示す．なお，異常検知型 AED は組織 A のログを用いている．

図より，異常検知型 AED，分散異常検知型 AED のいずれのケースでも，WL 型 AED と比較し CAPTCHA 発生回数を大幅に削減できていることが分かる．WL 型に近い検知率である，検知率 99%となるように設定した場合でも，日々の CAPTCHA 発生回数が 1 日平均で 6 回以下となることが確認できた．これはユーザ 1 人あたり 1 日 1 回未満となる．先行研究 [8] ではすべてのユーザが 1 日 1 回以下であれば CAPTCHA 発生が許容できるとしており，ユーザが許容できる利便性を維持できていることが分かる．なお，一部のユーザに集中して CAPTCHA が発生

表 7 CAPTCHA 発生数の統計
Table 7 Statistics of CAPTCHA occurrences.

AED 種別	合計	平均	最大	最小
WL 型	227	17.462	41	1
異常検知型 (TPR > 99%)	73	5.615	11	0
分散異常検知型 (TPR > 99%)	78	6	13	1
異常検知型 (TPR > 95%)	46	3.538	7	0
分散異常検知型 (TPR > 95%)	41	3.154	7	0
異常検知型 (TPR > 90%)	44	3.385	7	0
分散異常検知型 (TPR > 90%)	26	2	5	0

している可能性はあるが，その場合その他多くのユーザの CAPTCHA 発生数が少ないことを意味し，多くのユーザが利便性の低下を感じない点で効果があると考えられる．

また，検知率が 95%以下のケースでは，分散異常検知型 AED の方が異常検知型 AED と比較して CAPTCHA 発生回数を低減できていることが分かる．ユーザの利便性を重視したい環境であれば，分散異常検知型 AED を用いることにより比較的高い検知率を維持しながら，誤検知率を削減することが可能である．

4.4.3 処理性能

図 7 に分散異常検知型 AED を利用した場合の処理時間ごとの度数と累積比率を示す．処理時間の測定は，認証 Proxy に記録されている ICAP 処理時間を利用した．これは，認証 Proxy から AED への処理を ICAP [25] を用いて行っており，ICAP 処理時間を計測することで AED に関連する処理時間の測定が可能であると考えたためである．なお，図の x 軸が 9 秒の度数は，9 秒以降の値をすべて集計している．

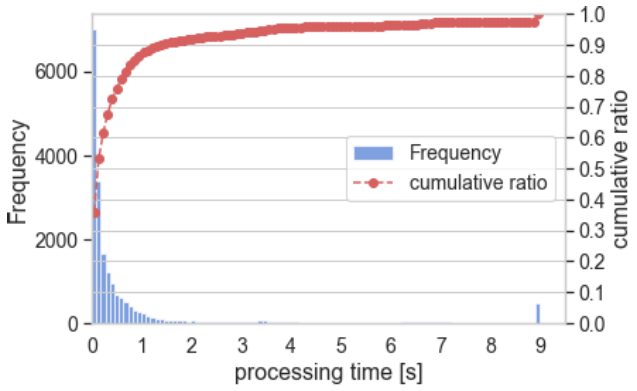


図 7 処理時間ごとの度数と累積比率

Fig. 7 Frequency and cumulative ratio by processing time.

図から 93.6%の処理時間が 3 秒以内であることが分かる．多くのユーザが Web ページのロードに期待している時間が 3 秒以内という調査結果 [26] があり，提案システムが実用の範囲内であることが分かる．

5. 関連研究

悪性サイトへの機械的な接続を遮断する方法として，インターネットへの出口に設置したプロキシのユーザ認証機能を利用する方法がある [27]．しかし，遠隔操作型マルウェアのなかにはユーザ認証機能を突破するものも存在している [28]．一方提案手法は，悪性サイトへの接続時に機械には突破困難な追加認証を課すことで対応している．

追加認証による対策は，日々の業務の妨げになるという課題が存在する．悪性サイトへの接続を防ぎつつも，安全性の高いサイトへの接続時には CAPTCHA を省略することで業務への影響を軽減する手法が角田らにより提案されている [29]．また，中鉢ら [30] は複数組織がブロックチェーン技術を用いて WL を共有することにより，AED が抱えていた，小規模組織で WL の拡張が小さく，追加認証が多く発生する課題を解決しようとしている．これらいずれの対策も，未知サイトへ接続する際に必ず追加認証が発生する．提案手法では，接続傾向を元に良性悪性の判定を行うことで，未知サイト接続時の追加認証の発生数を削減する．

未知サイトの良性悪性判定についても様々な研究が行われている．文献 [31] は，Convolutional Neural Network を拡張することで，プロキシログに含まれる宛先 URL の系列から，ドライブバイダウンロード攻撃に関する悪性 URL 系列を検知する手法を提案している．文献 [32] は，Bayesian sets と呼ばれる類似要素探索アルゴリズムを用いて，既知の悪性 URL 群と類似した URL を検索する方法を提案している．文献 [33] は，DNS ログからドメインと IP アドレスの関連をグラフ化し，確率伝播アルゴリズムを改良することで，悪性ドメインを高精度で検知する手法を提案している．これらの手法では，悪性サイトの判定にあらかじめ用意した悪性サイトのデータを必要とする．一方提案手法は，インターネットへの接続ログと，接続先 IP アドレスの AS 番号，国番号のみを使用し，悪性サイトのデータを必要としない点で異なる．

また，セキュリティ情報の共有に関しては，たとえば AlienValt OTX [34] では Indicator of Compromise（IOC）の共有が行われている．一方提案手法では，他組織が保有する接続ログといった，IOC ではない生データの情報を活用する点で異なる．

6. おわりに

本稿では悪性サイトへの通信遮断を目的として，(1) 組織が保有する，サイトへの接続ログを元に接続傾向を分析し，その傾向との乖離度を利用して接続の良性，悪性を判定する，(2) 自組織の接続傾向だけでなく，他組織の接続傾向を利用することで判定の精度向上を狙う，(3) 判定で誤って悪性と判断した良性サイトへの接続を即時遮断するのではなく，機械には突破困難な追加認証を課し，突破できなかった場合のみ接続を遮断する，という特徴を持つ分散異常検知型 AED というシステムを設計，提案した．

評価では，他組織の接続傾向を用いることによる，不審度検知の精度向上を示した．また，従来方式である WL 型の AED と比較して，異常検知方式を取り入れることにより，99%以上の検知率を維持した状態で，日々の CAPTCHA 発生回数を約 3 分の 1 に減らせることを示した．さらに，検知率 90，95%以上という，ある程度検知率の低下を許容したケースでは，自組織のみの接続傾向を用いる場合と比較し，他組織の接続傾向を用いることにより CAPTCHA 発生回数をさらに削減できることを示した．また，処理性能に関して，処理時間の 93.6%が 3 秒以内であることから，実運用に適応可能だと示した．

今後の課題としては，情報連携組織の拡大によるさらなる精度向上や，大規模なログを利用した長期運用がある．

組織をまたがったセキュリティ情報共有の重要性は論をまたないが，ログの機微性やデータ量の問題が，共有を活性化するうえでの障害となっている．また共有の効果を定量的に評価した研究は少ない．本研究は上記課題の解決を目指し，共有の効果を定量的に示したものであり，今後のセキュリティ情報共有の活性化の一助になれば幸いである．

参考文献

[1] Norton: What Are Malicious Websites?, available from ⟨https://nz.norton.com/internetsecurity-malware-what-are-malicious-websites.html⟩ (accessed 2021-02).

[2] Zhao, B.Z.H., Ikram, M., Asghar, H., Kaafar, M.A., Chaabane, A. and Thilakarathna, K.: A decade of malactivity reporting: A retrospective analysis of internet malicious activity blacklists, *Proc. 14th ACM Asia Computer Communication and Security (ASIA CCS'19)*, pp.1–13 (2019).

[3] IPA：「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，入手先 ⟨https://www.ipa.go.jp/files/000017308.pdf⟩（参照 2021-02）．
[4] 近藤賢郎，細川達己，重本倫宏，藤井康広，中村 修：分散型 SOC アーキテクチャに基づいた複数組織間におけるセキュリティ・オペレーションの連携，マルチメディア，分散協調とモバイルシンポジウム 2018 論文集，pp.872–878 (2018).
[5] 西嶋克哉，川口信隆，重本倫宏，近藤賢郎，中村 修：セキュリティ・オペレーションにおける秘匿データ分析システムの提案，マルチメディア，分散協調とモバイルシンポジウム 2020 論文集，pp.284–289 (2020).
[6] 仲小路博史，藤井康広，磯部義明，重本倫宏，鬼頭哲郎，川口信隆，林 直樹，下間直樹，菊池浩明：人間行動を用いた自律進化型防御システムの提案，2016 年暗号と情報セキュリティシンポジウム（SCIS2016），pp.1–8 (2016).
[7] Nakakoji, H., Fujii, Y., Isobe, Y., Shigemoto, T., Kito, T., Hayashi, N., Kawaguchi, N., Shimotsuma, N. and Kikuchi, H.: Proposal and Evaluation of Cyber Defense System Using Blacklist Refined Based on Authentication Results, *The 19th International Conference on Network-Based Information Systems* (*NBiS2016*), pp.135–139 (2016).
[8] 重本倫宏，藤井翔太，来間一郎，鬼頭哲郎，仲小路博史，藤井康広，菊池浩明：WL を用いた自律進化型防御システムの開発，情報処理学会論文誌，Vol.59, No.3, pp.1050–1060 (2018).
[9] Diffie, W. and Hellman, M.E.: New Directions in Cryptography, *IEEE Transactions on Information Theory*, Vol.IT-22, No.6, pp.644–654 (1976).
[10] Zhao, H., Chang, Z., Bao, G. and Zeng, X.: Malicious Domain Names Detection Algorithm Based on N-Gram, Hindawi, pp.1–9 (2019).
[11] Alina, O., Zhou, L., Robin, N. and Kevin, B.: MADE: Security Analytics for Enterprise Threat Detection, *Proc. ACM ACSAC'18*, pp.124–136 (2018).
[12] Michael, W. and Jun, W.: Unsupervised Clustering for Identification of Malicious Domain Campaigns, *Proc. ACM RESEC'18*, pp.33–39 (2018).
[13] Graham, C. and Muthukrishnan, S.: An improved data stream summary: The count-min sketch and its applications, *Journal of Algorithms*, Vol.55, No.1, pp.58–75 (2005).
[14] Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors, *Comm. ACM*, Vol.13, pp.422–426 (1970).
[15] GeoIP, available from ⟨https://dev.maxmind.com/geoip/⟩ (accessed 2021-02).
[16] Ayman, T. and Ali, S.H.: Anomaly Detection Methods for Categorical Data: A Review, *ACM Computing Surveys*, Vol.52, No.38 (2019).
[17] Jayaram, R. and David, J.M.: Unsupervised, low latency anomaly detection of algorithmically generated domain names by generative probabilistic modeling, *Journal of Advanced Research*, pp.423–433 (2014).
[18] Gao, H., Yan, J., Cao, F., Zhang, Z., Lei, L., Tang, M., Zhang, P., Zhou, X., Wang, X. and Li, J.: A Simple Generic Attack on Text Captchas, *23rd Network and Distributed System Security Symposium* (*NDSS '16*) (2016).
[19] Guixin, Y., Zhanyong, T., Dingyi, F., Zhanxing, Z., Yansong, F., Pengfei, X., Xiaojiang, C. and Wang, Z.: Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach, *Proc. 2018 ACM-SIGSAC Conference on Computer and Communications Security* (*CCS '18*), pp.332–348 (2018).
[20] Welcome to Flask — Flask Documentation (1.1.x), available from ⟨https://flask.palletsprojects.com/en/1.1.x/⟩ (accessed 2021-02).
[21] Empowering App Development for Developers | Docker, available from ⟨https://www.docker.com/⟩ (accessed 2021-02).
[22] PHISHTANK, available from ⟨https://phishtank.org⟩ (accessed 2021-02).
[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E.: Scikit-learn: Machine Learning in Python, *JMLR*, Vol.12, pp.2825–2830 (2011).
[24] Hunter, J.D.: Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, Vol.9, No.3, pp.90–95 (2007).
[25] RFC 3507 - Internet Content Adaptation Protocol (ICAP), available from ⟨https://tools.ietf.org/html/rfc3507⟩ (accessed 2021-02).
[26] SMARTBEAR: The Cost of Poor Web Performance, available from ⟨https://smartbear.com/blog/test-and-monitor/the-cost-of-poor-web-performance-infographic/⟩ (accessed 2021-02).
[27] IPA：「高度標的型攻撃」対策に向けたシステム設計ガイド，入手先 ⟨https://www.ipa.go.jp/files/000046236.pdf⟩（参照 2020-02）．
[28] IIJ：新型 PlugX の出現，入手先 ⟨https://sect.iij.ad.jp/d/2013/11/197093.html⟩（参照 2020-02）．
[29] 角田 朋，大鳥朋哉，藤井康広，谷口信彦，木城武康：グレーリストを用いたホワイトリスト／ブラックリストの自動生成によるマルウェア感染検知方法の検討，情報処理学会研究報告，SPT, Vol.2014, No.16, pp.1–7 (2014).
[30] 中鉢かける，中村嘉隆，稲村 浩：標的型攻撃対策のためのブロックチェーン技術を用いたホワイトリスト方式防御システムの実現性に関する検討，研究報告モバイルコンピューティングとパーベイシブシステム（MBL），Vol.2018-MBL-89, No.6, pp.1–8 (2018).
[31] 山西宏平，芝原俊樹，高田雄太，千葉大紀，秋山満昭，八木毅，大下裕一，村田 正：畳み込みニューラルネットワークを用いた URL 系列に基づくドライブバイダウンロード攻撃検知，コンピュータセキュリティシンポジウム 2016 論文集，pp.811–818 (2016).
[32] 孫 博，秋山満昭，八木 毅，森 達哉：既知の悪性 URL 群と類似した特徴を持つ URL の検索，コンピュータセキュリティシンポジウム 2014 論文集，pp.1–8 (2014).
[33] Hau, T., An, N., Phuong, V. and Tu, V.: DNS graph mining for malicious domain detection, *2017 IEEE International Conference on Big Data* (*Big Data*), pp.4680–4685 (2017).
[34] AlienVault - Open Threat Exchange, available from ⟨https://otx.alienvault.com/⟩ (accessed 2021-02).

西嶋 克哉

2014年早稲田大学先進理工学研究科電気・情報生命専攻修士課程修了．同年ITベンダーに入社しエンジニアとして従事．2018年7月より株式会社日立製作所にてサイバーセキュリティ，情報セキュリティの研究開発に従事．CISSP．

川口 信隆 （正会員）

2008年3月慶應義塾大学大学院理工学研究科開放環境科学専攻後期博士課程修了，博士（工学）．同年4月より株式会社日立製作所にてサイバーセキュリティ，情報セキュリティの研究開発に従事．コンピュータセキュリティ研究会専門委員，電子情報通信学会論文誌編集委員．IEEE各会員．CISSP．

植木 優輝

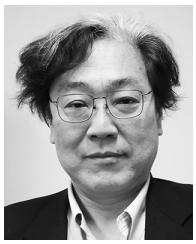
2020年3月東京農工大学大学院工学府物理システム工学専攻前期博士課程修了．同年4月より，株式会社日立製作所にてサイバーセキュリティ，情報セキュリティの研究開発に従事．

重本 倫宏 （正会員）

2006年大阪大学大学院基礎工学研究科システム創生専攻修士課程修了．同年（株）日立製作所システム開発研究所（現，システムイノベーションセンタ）入所．現在はネットワークセキュリティ技術に関する研究開発に従事．博士（工学）．

近藤 賢郎 （正会員）

2015年慶應義塾大学大学院理工学研究科修士課程修了．2016年同大学大学院医学研究科修士課程修了．同年より同大学大学院理工学研究科博士課程．2013～2017年まで同大学大学院理工学研究科研究員．2017年4月～2020年10月まで同大学インフォメーションテクノロジーセンター本部助教．2020年11月同大学情報セキュリティインシデント対応チーム助教．2017年4月同大学サイバーセキュリティ研究センター所員（兼務）．ネットワーク・アーキテクチャ，大規模コンテンツ配信基盤，セキュリティ運用技術，認証認可に関する研究に従事．電子情報通信学会，IEEE，ACM各会員．WIDEプロジェクトボードメンバ．



中村 修

1983年慶應義塾大学工学部数理工学科卒業，1990年大学博士課程単位取得退学．博士（工学，慶應1993年）．1990年から東京大学大型計算機センター助手を経て1993年慶應義塾大学環境情報学部助手となり，現在，慶應義塾大学環境情報学部教授．1987年からWIDEプロジェクトにてインターネットの研究開発に携わる．広帯域インターネットやIPv6の研究開発，普及に携わる．2009年から藤沢地域のWiMAXの運用会社オープンワイヤレスプラットフォーム合同会社の技術協議会委員長となり，無線インフラを含めたインターネット関連の研究開発も行う．2014年から4年間W3C/KEIO Site Managerとして，Web技術の標準化活動を推進する．2017年から慶應義塾インフォメーションテクノロジーセンター所長．ACM，電気情報通信学会，ISOC各会員．WIDEプロジェクトボードメンバ．