TLS暗号化通信中の悪性通信識別について

佐藤 龍[1,2,a)]　土井 洋[1,b)]

概要：セキュリティオペレーションセンタ (SOC) は，さまざまなセキュリティ製品を活用して組織内の脅威を検出する役割を担っている．その中でも，WAF，IPS，IDS，UTM 等の多くの製品が通信ペイロード分析する手法を採用しており，SOC の分析はペイロードに依存している．しかし近年，TLS による暗号化通信の割合が急増し，従来の通信ペイロードによる分析が困難となっている．このような背景から，セキュリティ製品には TLS 通信を途中で復号する TLS インスペクション機能が備えられるようになったものの，導入の手間や CA 証明書の追加に非対応の IoT 機器などが障壁となり，TLS インスペクション機能の活用は完全とは言えないのが現状である．そこで本稿では，TLS インスペクションを用いることなく暗号化通信中の悪性通信を識別する手法について調査を行い，TLS1.3 プロトコルを含む最新の暗号化通信環境で懸念される影響について，実験を通じてその影響度合いを見積もり新たな課題として提起する．

キーワード：SOC，SSL，TLS，マルウェア検知

Detection of malicious traffic in TLS encrypted traffic.

Sato Ryu[1,2,a)]　Doi Hiroshi[1,b)]

Abstract: Security operation centers (SOCs) are responsible for detecting threats in organizations by utilizing various security products. Among them, many products such as WAF, IPS, IDS, UTM, etc. employ communication payload analysis methods, and SOC analysis relies on the payload. In recent years, however, the ratio of TLS-encrypted communications has increased rapidly, making it difficult to analyze conventional communication payloads. Against this backdrop, security products are now equipped with a TLS inspection function that decrypts TLS communications in progress. However, the use of the TLS inspection function is not yet complete due to barriers such as the time and effort required for installation and IoT devices that do not support the addition of CA certificates. In this paper, we investigate a method to identify malicious communication during encrypted communication without using TLS inspection, and estimate its impact on the latest encrypted communication environment including TLS1.3 protocol through experiments, and raise it as a new issue.

Keywords: Malware Detection，SOC，SSL，TLS

1. 序論

1.1 TLS の普及

近年の情報セキュリティ意識の高まりから，TLS による常時暗号化が急速に普及しており，2025 年 04 月現在で 81%以上の Web 閲覧トラヒックが TLS により暗号化されていることが，Google 透明性レポート [1] によって報告されている．悪性通信に着目すると，2021 年時点で 46%のマルウェアが C2 通信を TLS によって暗号化していたとの調査結果を Sophos が報告している [2]．また Oh らは，TLS によって通信が暗号化された場合，従来 SOC で用いていたペイロード分析が行えなくなり，SOC でのネットワークトラヒック分析に深刻な影響を与えると主張する [3].

[1] 情報セキュリティ大学院大学 (Institute of Information Security)
[2] 株式会社エヌ・ティ・ティ エムイー (NTT-ME)
[a)] mgs245506@iisec.ac.jp
[b)] doi@iisec.ac.jp

1.2 TLS インスペクションの課題

TLS による通信の暗号化に対応するため，SOC では TLS インスペクションによって暗号化された通信のペイロードを分析している．しかし，ネットワーク内の全端末に対して設定が必要となる点や，互換性がない機器やソフトウェアが存在する点が導入の障壁となり，結果として，TLS インスペクションが無効化されることもしばしばあり，悪性通信の見逃しが懸念されている．

1.3 TLS1.3 の台頭

2018 年に RFC8446 として規格策定された TLS1.3 もまた，急速に普及しており，2024 年 05 月時点で普及率 70.1%という調査結果が Qualys SSL Labs によって報告されている [4]．加えて，本稿 4 章にて述べる独自の調査では 2025 年 04 月時点で TLS1.3 の普及率が 82%まで上昇していることが確認できた．

1.4 TLS1.3 データセット不足の課題

TLS1.3 の普及という背景を受け，悪性通信検出への影響を明らかにすることが喫緊の課題となっている．しかし，TLS1.3 通信の公開データセットは限られており，悪性通信データセットについては知る限りでは存在しない．このような背景から，TLS1.3 が悪性通信検出へ与える影響を実験的に明らかにした研究は極めて少なくなっている．

1.5 目的

TSL1.3 にはプライバシー強化などのメリットがある一方，既存の悪性通信識別手法に対して影響を与える可能性がある．これについて Barradas(2024) らは，以下のような点が悪影響を与えると警鐘を鳴らしている [5]．

- ハンドシェイクが一新され，クライアントとサーバ間でやりとりされるレコードのシーケンスが変更された
- もはや証明書の情報は分類に利用できなくなった
- 外部の観測者からはプロトコル関連の情報とアプリケーションの情報が明確に分離できなくなった

そこで本稿では，TLS 暗号化通信によって生じる悪性通信を分析する上で活用可能なペイロード情報の欠落という課題の解決を目指し，TLS 暗号化トラヒック中の悪性通信を識別する手法の特徴や課題について調査し，特に最新の TLS1.3 プロトコル環境下での有効性を実験的に明らかにする．

1.6 貢献

本稿では以下の研究課題解決を貢献とすることを目指す．

研究課題: 先行研究の有効性が TLS1.3 環境下でどのように変化するかを明らかにする

TLS 暗号化通信中の悪性通信識別に焦点を当てた先行研究は多数存在するが，TLS1.3 を対象としていることが明確に述べられた研究は少ない．さらに，TLS1.3 環境下での評価実験も知る限りでは存在せず，SOC が運用する上での判断材料が存在しない．そのため，本研究では先行研究の有効性が TLS1.3 環境下でどのように変化するかを比較検証して明らかにする．

2. 関連研究

2.1 ネットワークトラヒック分析 (NTA)

Oh(2021) らは，ネットワークトラヒック分析 (NTA) を，トラヒックをネットワークトレースやフロー記録の形式で保存し，エンドホスト，ユーザー，アプリケーション，プロトコルを推論することとしている [3]．NTA は利用する情報によって二種類に分けられ，CISCO Netflow に代表されるメタデータを用いる方式と，アプリケーション層のペイロードを用いる方式が存在する．SOC では，WAF や IPS といったセキュリティアプライアンスがペイロードを利用しており，欠かすことのできない要素となっている．しかし，これら従来の方式では暗号化されたペイロードを分析する事はできない．

2.2 TLS インスペクション

TLS インスペクションは，従来の NTA 方式の課題であった TLS 通信の分析ができない点を解決する．

TLS インスペクションの構成を図 1 に示す．この手法では，クライアントとサーバ間にミドルボックス[*1]を設置する．クライアントが TLS 通信を開始すると，ミドルボックスは，クライアント・ミドルボックス間，ミドルボックス・サーバ間で別々に TLS のセキュアチャネルを構築する．これにより，ミドルボックスがセキュアチャネルの終端となり，平文のペイロードを得て分析することが可能となる．

この手法では，ペイロードを従来のシグネチャで分析可能であり，運用ノウハウが再利用可能という利点がある一方で，エンドツーエンド暗号化を破壊するという課題がある．エンドツーエンド暗号化の破壊は，クライアントが中間者攻撃を検知して通信を切断するという動作を引き起こすため，クライアントに対して，ミドルボックスの電子証明書を「信頼できる証明書発行機関」に登録する事前設定が必要となる．この設定が行えない機器やアプリケーションには，TLS インスペクションを利用できない．

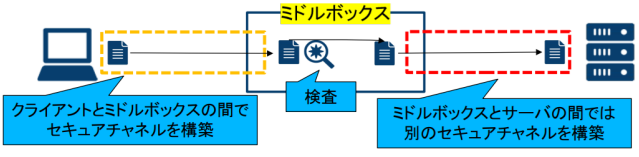

図 1 TLS インスペクションの構成

[*1] 通信の途中に割り込む事ができるプロキシ等の装置

2.3 セッション鍵共有

セッション鍵共有はエンドツーエンド暗号化を破壊することなく，TLS インスペクションと同様にペイロードを得ることができる．

セッション鍵共有の構成は図 2 に示す通り，サーバとクライアント間の通信を複製して傍受可能な箇所にミドルボックスを設置する形となる．この手法では，サーバが用いる電子証明書および秘密鍵をミドルボックスと共有する．ミドルボックスはこの鍵ペアを用いてセッション鍵を抽出し，さらに TLS 通信を復号して平文のペイロードを取得・分析することが可能となる．サーバの鍵ペアをミドルボックスと共有するという制約があるため，主に WAF などのサーバ向けのアプライアンスで用いられる事が多い手法である．

この手法の課題としては，Cipher Suite が RSA 暗号ベースに限定されてしまう点が挙げられる．従って，前方秘匿性を持たない暗号が削除されている TLS1.3 では，この方式は利用できない．

この課題を解決するため，サーバ側の乱数を固定した Static DH によって DH 暗号ベースの Cipher Suite でもセッション鍵共有を利用する手法が提案されている [6]．しかし，Static DH はセキュリティ上の懸念から利用を控えるよう呼びかけられている [7]．

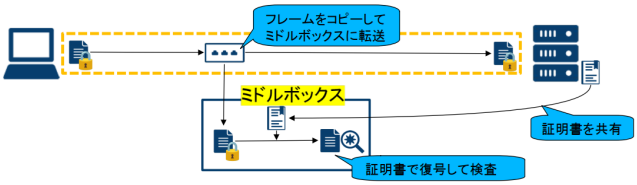

図 2 セッション鍵共有の構成

2.4 静的分析

静的分析は，パケットヘッダなどの情報を利用して通信の内容を推定する手法である．ペイロードを必要としないため，軽量に動作し暗号化の有無に影響されない．

Velan(2015) らは粒度と結果の安定性の関連について検証を行った．この検証では，「バイト単位」「パケット単位」「フロー単位」の 3 つの粒度での実験を行い，粒度が大きい「フロー単位」の結果が最も安定していることを示した [8]．

Papadogiannaki(2021) らはパケット長の並びをシグネチャとして用いる Header Hunter を提案し，暗号化されたトラヒックを安価なハードウェアで高速に分析できることを示した [9]．

2.5 TLS 指紋

TLS 指紋とは，TLS 通信を行うサーバやクライアントを識別するために，通信パラメータから生成される識別子である．この指紋の生成には，表 1 に示すような，TLS ハンドシェイクで交換される暗号化されていない情報を利用する．また，TLS 指紋がサーバやクライアントを識別できるという特徴から，悪性通信の IoC として利用されている．

表 1 TLS 指紋生成に用いられる情報

Client	TLS バージョン，Cipher Suites， SNI，TLS バージョン， クライアント証明書
Server	TLS バージョン，Cipher Suite， SNI，TLS バージョン， ALPN(上位プロトコル)，サーバ証明書

2.6 統計的分析

統計的分析は，ネットワークトラヒックにおけるパケット長や到着間隔といった統計的特徴を分析し，マルウェア特有の通信パターンや振る舞いを検出する手法である．この手法は機械学習との親和性が高く，例えば Korczyński らのマルコフモデルを用いる手法 (2014)[10] や，Bartos らの度数分布図と自己相似配列を用いる手法 (2016)[11] など，近年になって多様な研究が行われている．

3. 準備

3.1 ネットワークフロー

ネットワークフローは，ネットワーク上の通信を可視化するための単位である．狭義には Netflow や IPFIX[12] といった，トラヒックを収集するためのプロトコルによって生成されるデータを指す．しかし現在では，これら特定のプロトコルに限定されず，共通の特性を持つパケットを集約した単位やそれらの統計情報を指すことが一般的である．そのため，「タプル」と「方向」の 2 要素を調整し，ネットワークフローの集約方法や粒度を定義する．

タプル

タプルとは，個々のパケットを同一フローとして集約する際にキーとなるパラメータの組み合わせを指す．本稿では以下の「5 タプル」および「4 タプル」を使用している．

5 タプル: 送信元，送信元ポート，宛先，宛先ポート，プロトコル

4 タプル: 送信元，宛先，宛先ポート，プロトコル

方向

通信の方向性に関する集約方法として、「単一方向フロー」と「双方向フロー」の 2 種類が存在する。本稿では，「双方向」を用い，関連する行きと帰りのパケットはどちらも同一のフローとして扱う．

フローはパケット情報よりも集約されているため，端末やアプリケーションの挙動分析に適す．この特性が様々な目的で応用された結果多くのバリエーションが存在しており，例えば，McGrew(2016) らは TLS 通信を復号すること

なく分析するために必要な情報を集約した，強化 TLS フロー記録 [13] を提案している．

3.2 JA3

JA3 は 2017 年に John Althouse らが開発した TLS 指紋の生成手法であり，Salesforce, Inc. からオープンソースで提供されている [14]．また，サーバの指紋を生成する「JA3S」と呼ばれる派生版も存在する．

指紋生成の過程では，はじめに，TLS 通信の Client Hello メッセージから「バージョン」「暗号スイート一覧」「拡張一覧」「楕円曲線一覧」「楕円曲線ポイントフォーマット一覧」の順で抽出する．続いて，これらのバイト列をハイフン区切りの decimal 表現に変換．さらに順番通りにカンマ区切りで結合して「769,4–5–10–9–100–98–3–6–19–18–99,,,」のような文字列を得る．最後に，この文字列の MD5 メッセージダイジェストを生成し，これを指紋として用いる．JA3S では，はじめに Server Hello メッセージから「バージョン」「暗号スイート一覧」「Extension タイプ一覧」の順で抽出する点以外，指紋生成方法は共通する．

JA3 や JA3S は現在主流の TLS 指紋生成方式となっており，AWS WAF など大手の製品に導入されているほか，現場での利用例も存在する．例えば，小林らが SOC の現場にて DDoS 攻撃の検知および防御に JA3 を用いた事例を 2024 年に報告している [15]．

3.3 MVDet

3.3.1 概要

MVDet[16] は，2024 年に Cui らが提案したマルチビュー分析に基づくマルウェア検出手法である．マルチビュー分析は複数の情報を横断的に利用するため，個々の情報源が持つ利点の相乗効果を期待できる．

3.3.2 選定理由

Cui らは本研究と同様に TLS 暗号化されたマルウェア通信の検出という課題に取り組み，その中でマルチビュー分析という手法を導入した研究を行っている．

マルチビュー分析に用いる情報源は「ビュー」と呼び，MVDet においては 4 種類を分析対象としており，MVDet 全体としての特徴もこれら 4 種類の「ビュー」の影響を受けると考えられる．そこで，ビューごとに分けて特徴の分析を実施した．

はじめに，4 種類の「ビュー」と，2 章「関連研究」で述べた手法との対応を以下に示す．

- **統計ビュー:** 2.6 節「統計的分析」
- **DNS ビュー:** 2.6 節「統計的分析」
- **TLS ビュー:** 2.5 節「TLS 指紋」
- **ビジネスビュー:** 2.1 節「NTA」

次に，2 章「関連研究」にて述べた 6 種類の手法が持つ特徴を，表 2 に示す．ここでは，比較の観点として「TLS 分析 (が可能)」「軽量[*2]」の 2 点，SOC 業務において実用的な観点から重要となる「低過検知率」「亜種耐性」の 2 点，および TLS インスペクションの導入障壁となっている「透過的[*3]」の計 5 点を比較した．

これらのうち MVDet は「統計的分析」「TLS 指紋」「NTA」に関連するビューを組み合わせた分析を実施している．これにより，本研究における「TLS 分析 (が可能)」「透過的」という前提条件を満たす．さらに Cui らによって「低過検知率」と「亜種耐性」を両立することが示唆されている点から [16]，本研究において重視している 5 つの観点をすべて満たす可能性のある MVDet を重要な先行研究と位置付けた．

3.3.3 統計ビュー

Cui らは，攻撃ホストと被害ホストとの間に発生するフローには類似した傾向があると述べている [16]．特に，表 3 に示す 11 種類は重要な統計的特徴量と位置づけられ，マルウェア検出に有効であるため，MVDet においてもこれら 11 種類の特徴量を「統計ビュー」として用いている．

3.3.4 DNS ビュー

マルウェアは C2 通信自体を DNS 通信に偽装する目的と，C2 サーバの IP アドレスを名前解決する目的の二種類の目的で DNS 通信を行う場合がある．前者は統計ビューでの検出が期待できる．しかし，後者ではこれが期待できないため，DNS レコードに着目する．C2 サーバの DNS レコードに設定される TTL は通常の DNS レコードと比較して短い傾向があり，これが正常な通信と C2 通信の名前解決を見分ける上での参考となる [16]．

このような傾向をマルウェア検出に役立てるため，MVDet においては表 4 に示す 5 種類の特徴量を「DNS ビュー」として抽出し，ネットワークフローと紐付けて利用する．

3.3.5 TLS ビュー

ネットワークフローが TLS 通信を含む場合，3.2 節の JA3 による TLS 指紋を生成し，これを「TLS ビュー」として用いる (表 5)．

3.3.6 ビジネスビュー

ビジネスビューとは，ネットワークフローが関連するサービスやアプリケーションを識別する事を指す．これはマルウェアの活動とも強い相関性があり，例えば，マルウェアが C2 サーバに対してファイル転送を行うフローはファイル転送サービスと関連している．対照的に，インスタントメッセージやストリーミングといったサービスはマルウェアの活動との関連性が低い．Cui らは，このような傾向を分析し，悪性通信が「Web」「ダウンロード」のサービスと，「未知」のアプリケーション分類に集中しており，正常な通信とは傾向が異なることを示した [16]．

MVDet では nDPI[17] を用いて表 6 の「ビジネスビュー」

[*2] 分析にペイロードを用いない性質
[*3] TLS のエンドツーエンド暗号化を破壊しないという性質

表 2　関連研究の特徴比較

特徴/手法	NTA	TLS インスペクション	セッション鍵共有	静的分析	TLS 指紋	統計的分析
TLS 分析		✓	✓	✓	✓	✓
軽量				✓	✓	✓
低過検知率	✓	✓	✓			
亜種耐性						✓
透過的	✓		✓	✓	✓	✓

表 3　統計ビューの特徴量

特徴	種類	例
上り通信量	数値	62,134
下り通信量	数値	35,279
合計通信量	数値	97,413
上りパケット長の平均	数値	1,479.38
下りパケット長の平均	数値	1,469.96
パケット長の平均	数値	1,475.95
合計パケット数	数値	66
フローの間隔	数値	330
フローの到着間隔	数値	5
送信元ポート	数値	43,760
宛先ポート	数値	443

表 4　DNS ビューの特徴量

特徴	種類	例
クエリ	文字列	www.iisec.ac.jp
クラス	文字列	IN
タイプ	文字列	A
サブクエリ	文字列	-
TTL	数値	86,400

表 5　TLS ビューの特徴量

特徴	種類	例
JA3	文字列	4d7a28d6f2263ed61de88ca66eb011e3
JA3S	文字列	80b3a14bccc8598a1f3bbe83e71f735f

を判定する．これにより効果的なマルウェア通信の検出が行えたとする一方で，nDPI よりも高度な手法があればMVDet の有効性が増すとも述べている [16].

表 6　ビジネスビューの特徴量

特徴	種類	例
サービス	文字列	Google
アプリケーション	文字列	Web 閲覧

3.3.7　4 タプルへの集約

マルウェア検出の場面においては，5 タプルフローと比較して 4 タプルフローがより頑強である [16]．そのため，5 タプルフローに紐付けて収集された「統計」「DNS」「TLS」「ビジネス」の 4 つのビューを 4 タプルフローに集約する．

集約の過程では，5 タプルフローの特徴量に対して統計処理を行い，同じ 4 タプルフローに含まれる 5 タプルフロー間の特徴量を包括的に表す．MVDet における各ビューの特徴量は「数値」と「文字列」に分けられる．数値の特徴量は，最大，最小，平均，分散を計算する．文字列の場合は，用語数，最頻出項目，その比率を計算する．

3.3.8　学習

Cui らは複数の機械学習フレームワークを比較し，マルウェア通信と正常通信を二項分類する場面においては，LightGBM[18] が最も効果的な機械学習フレームワークであると結論づけた [16].

LightGBM は並列学習をサポートするため学習が高速であり，低メモリで動作し，高精度である点など，多くのメリットを持つ．そのため Cui らは，大量のトラヒックを高い応答速度で処理するために LightGBM を推奨している．

3.3.9　MVDet の性能評価

CSE-CIC-IDS2018 のデータセットを用いて実施した評価は次の通りである．Cui らによると，現在一般的に利用されている Kitsune や Joy などの手法と比較して，全攻撃種に対して最も高い真陽性率であるなど，MVDet が高い性能を持つ事が示されたとしている [16].

表 7　MVDet の性能評価 (CSE-CIC-IDS2018, 閾値 FPR=0.0236)

種類	真陽性率	偽陽性率	AUC
Bot	0.9088	0.0135	0.9639
DoS	0.9844	0.0095	0.9893
DDoS	0.9801	0.0120	0.9970
総当たり	0.9571	0.0236	0.9877
SQLi	0.9610	0.0091	0.9932
侵入	0.9426	0.0084	0.9801
平均	0.9557	0.0127	0.9852

3.3.10　MVDet における課題

MVDet はビジネスビューを含んだマルチビュー分析を初めて提案し，性能評価においても非常に高い性能を発揮している．しかし，次のような最新プロトコルを考慮した設計ではなく，最新の環境では性能が未知数である．

- DNS over TLS
- TLS1.3
- QUIC
- HTTP/3

なかでも TLS1.3 は普及率が極めて高く，使用回避が難しい．さらに TLS1.3 が QUIC や HTTP/3 の基礎技術である点も考慮し，本研究では TLS1.3 を対象としたマルチビュー分析モデルの性能に関する比較検証を実施する．

4. TLS ビューに対する TLS1.3 の影響調査

TLS1.3 環境が先行研究の手法に与える影響を正確に調査するためには，活動期間中に TLS1.2 から TLS1.3 へと移行したマルウェアのデータセットが必要となる．しかし，TLS1.3 通信に焦点を当てたマルウェアデータセットは知る限りでは存在しない (1.4 節).

そのため，まずはじめに TLS1.3 を含む正常な通信を実際に発生させ，TLS1.2 以前と TLS1.3 以降での TLS ビューの振る舞いや特徴の変化から，影響を考察することとした．

4.1 調査方法

TLS1.3 の正常通信は，Selenium で Firefox のブラウザ操作を自動化し URL リストを自動的に巡回して生成する．

4.1.1 URL リスト

URL リストは Chrome Top Million Websites(CrUX)[19] を加工したものを使用する．CrUX は，Google が収集した Chrome ブラウザの閲覧履歴を元に作成されており，アクセス数の多い上位 100 万件の Web サイトが含まれる．Ruth らの調査によると，Chrome から発生するトラヒックの 95%は上位 100 万件の Web サイトから生成されており，トラヒック全体を近似するための合理的な量とされる [20]．加えて，頻繁にリストが更新されているため，リスト内のリンク切れなどの影響も少なく本調査において理想的であると判断した．また，本研究では CrUX の 2025/02 のデータセットを利用し，うち 1 万件を無作為に抽出した．

4.1.2 トラヒックの収集

トラヒックの収集には tshark を用い，次のフィルタ設定を適用する．”tcp port 80 or tcp port 443 or udp port 53 or udp port 80 or udp port 443”

tshark で収集したトラヒックから JA4S 指紋[*4]を生成し，傾向の変化を分析する．

4.2 調査結果

調査結果を表 8 に示す．全 10,000URL 中，TLS 通信に対応した件数は 6,079 であった．その中でも TLS1.3 の件数は 5,011 であり，TLS 通信全体における TLS バージョンの内訳では TLS1.3 が 82%となった．これは，2024 年の調査 [4] と比較して 12 ポイント増加している．

続いて，一意の TLS 指紋数を集計したところ，TLS1.2 では 105 通り確認された指紋の種類が TLS1.3 では 6 通りまで減少した (以降，TLS1.3 で観測された TLS 指紋を TLS1.3 指紋と呼称する)．さらに，今回の調査で発見できた 6 種類の TLS1.3 指紋の出現回数の調査結果 (表 9) からは，頻度に大きな偏りがあり，主に観測される TLS1.3 指紋に着目すると 3 種類にまで数を減じることが示された．

[*4] JA4 は TLS バージョン情報などの可読性が高く分析に向くため選定．先行研究で用いられる JA3 とは設計に類似性がある．

これらの結果を総合すると，TLS 指紋は TLS1.3 環境においてはサーバーアプリケーションをほとんど区別できていないと考えられる．

表 8 TLS バージョンごとのサイト数および一意の指紋数

バージョン	URL 数	一意の指紋数
平文/応答なし	3,921	-
TLS1.2	1,068	105
TLS1.3	5,011	6

表 9 TLS1.3 通信における各 TLS 指紋の出現回数

指紋	出現回数
TLS 指紋 1	2354
TLS 指紋 2	1421
TLS 指紋 3	1202
TLS 指紋 4	18
TLS 指紋 5	10
TLS 指紋 6	6

4.3 考察

TLS1.3 においてサーバーアプリケーションを識別できない要因としてハンドシェイクプロセスの刷新が考えられる．これにより指紋生成に使われる「TLS バージョン」「Extensions」「Cipher Suite」が次のように変更された [21]．

- TLS バージョンを TLS1.2 と同じ「0x0303」に固定
- Extensions の大部分が暗号化．調査ではほぼ全ての通信が TLS1.3 を示す符号 (タイプ 43) と鍵交換 (タイプ 51) の二種類のみを用いていた．
- Cipher Suite は 5 種類まで減少．これが TLS 指紋を変化させるほぼすべての要素であった．

TLS1.3 において，TLS 指紋を変化させる大きな要素が 5 種類の Cipher Suite であるという点は，調査において一意の TLS 指紋の数が 6 件であった事とも一致する．

TLS1.3 環境における一意の指紋の数に着目した場合，クライアントの指紋は TLS バージョンの影響を受けにくいと考えられる一方で，サーバの指紋はサーバアプリケーションの識別を困難にするまでに一意の数が減少していた．そのため，MVDet においても両者の組み合わせの数が減少する事による悪影響を受ける可能性が高い．

5. TLS パラメタの揺らぎへの頑強性の調査

TLS 指紋を対象とした調査 (4 章) によって MVDet のようなマルチビュー分析モデルが TLS1.3 環境で悪影響を受ける可能性が高い事が示唆された．そのため，TLS ビューが悪影響を受けた場合の検出器全体としての頑強性を調査するため，MVDet と同様のパラメタを使用したマルチビュー分析モデルを作成し，TLS 関連パラメタに揺らぎを与える実験を行った．

TLS1.3 に移行することによる指紋の減少は，「TLS パラ

メタの揺らぎ」として一般化することができる．そのため，TLS パラメタに揺らぎが発生した場合のモデルの頑強性を観察することで，TLS1.3 に移行した際の影響度合いを見積もることが可能であると考えた．

5.1 実験方法

Wickramasinghe(2025) らは，TLS1.3 環境が暗号化悪性通信分類器に与える影響を調査する過程で，分類器の頑強性についての調査を実施している．その際に，過学習が与える影響という文脈で，学習時に過学習が発生しやすいパラメタを推論時に遮蔽 (obfuscate) することで頑強性をシミュレーションする手法を用いた [22]．彼らは遮蔽の方法を”Randomize (replace relevant bytes with $0xnn$, where n is a random hexadecimal value.)”としており，本稿でもその手法に倣って TLS パラメタの遮蔽を行う．

本研究では，データセットとして Malware Capture Facility Project により収集された「Malware Captures」および「Normal Captures」の pcap ファイルを用いる [23]．学習および評価には，それぞれ異なるサンプルを用いた．特にマルウェアに関しては，学習に用いたマルウェアと同一のファミリに属する別の検体（亜種）のみを評価に利用するように選定した．その概要を表 10 に示す．

表 10 データセット概要

	学習		評価	
	悪性 ‡	良性	悪性 ‡	良性
pcap ファイル	46	16	5	1
マルウェアファミリ	44	-	5	-
フロー	27,551	43,228	1,099	1,457
合計フロー	70,779		2,556	
重複排除後	54,130		1,611	

‡ 無関係の通信を手動で除外し悪性通信のみを抽出する加工を実施

5.2 評価手法

二項分類の性能を測る指標としては通常，TPR，FPR，F 値といった指標が利用される．しかし，これらの指標は閾値の影響を大きく受ける．SOC においても閾値を操作して TPR，FPR を組織の要求する値に調整することがしばしばあり，したがって，「閾値を 0.5 に固定する」「FPR を 0.0236 になるように閾値を設定する」といった状況での評価は実務視点からは不十分である．

これに対し，閾値の影響を受けない指標として ROC が用いられるが，クラス不均衡が生じる場合や少ないテストケースにおいてモデルの性能を正確に反映しない場合があり，今回の実験ではその可能性が高いと考えられる．そこで，今回は「悪性通信の判定結果の度数分布図」および「偽陰性率が 0 となる最大の閾値 (以降 Zero-FNR Threshold を略し ZFT と呼称)」を用い，TLS パラメタの揺らぎがモデルに与える影響を可視化することを試みた．

5.3 実験結果

著者が作成したマルチビュー分析モデルにて，評価用データセットの悪性通信フローを直接推論した場合と，TLS ビューを遮蔽した上で推論した場合の，フローごとの判定結果の度数分布をそれぞれ図 3，4 に示す．判定はフローごとに「マルウェアらしさ」の確率が 0 から 1 の間で出力される．そのため検出器が理想的である場合には，ZFT(偽陰性率が 0 となる最大の閾値) は限りなく 1 に近く，度数分布図は最も大きい値の階級にすべてのフローが収まる．

評価用データセットを直接推論した場合 (図 3)，ZFT が 0.990，度数分布図は最も大きい値の階級にすべてのフローが収まり，亜種の検出においても高い性能を発揮していた．一方，TLS ビューの遮蔽を行った場合 (図 4) は，ZFT が 0.008，度数分布図は全階級にわたって分散しており，TLS ビューの遮蔽が判定に影響を与えることが確認できた．

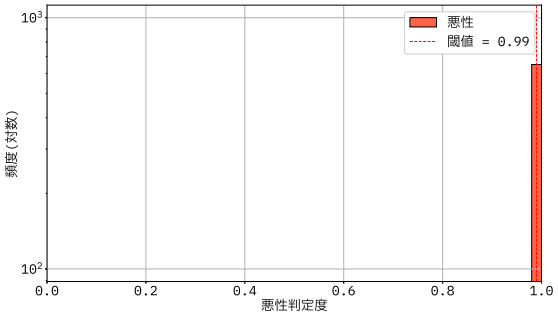

図 3 遮蔽を行っていない場合の度数分布図

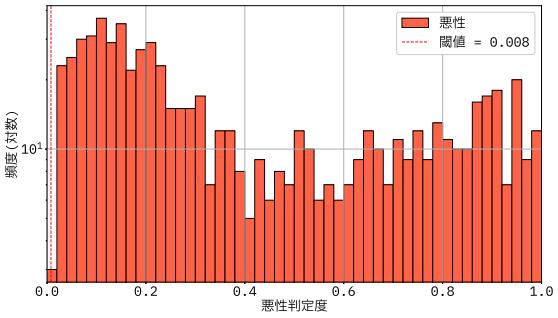

図 4 TLS 指紋を遮蔽した場合の度数分布図

5.4 考察

TLS パラメタの揺らぎが，悪性通信の検出を目的としたマルチビュー分析モデルの ZFT(偽陰性率が 0 となる最大の閾値) および判定結果の度数分布図上に大きな影響を与えることが確認され，TLS パラメタの揺らぎに対する頑強性の低さという課題が浮き彫りになった．

6. 結論

本稿では，TSL 暗号化通信における悪性通信をペイロード分析に頼らずに検出する手法について，TLS1.3 が与える影響に焦点を当てた調査を行った．

まず, 通常の Web サイトを対象とした調査により, TLS1.3 が 82%という高い普及率にあることを再確認した．その上で，TLS1.3 環境では TLS 指紋によってサーバを識別する能力が著しく低下することを実験的に明らかにした．

続いて，TLS 指紋の有効性低下がマルチビュー分析モデルに与える影響を見積もるため，評価データセットの TLS 指紋を意図的に遮蔽する実験を行った．その結果，TLS 指紋が利用できない場合に，モデルの悪性通信検出性能が大幅に劣化する可能性が示唆された．このような影響が生じる要因としては，TLS 指紋の生成過程に暗号学的ハッシュ関数が関与していることが挙げられる．ハッシュ関数の性質により小さな揺らぎが TLS 指紋全体へと増幅され，判定への影響を強めたと考えられる．

したがって，TLS パラメタの揺らぎに対する頑強性の低さという課題を克服するためには，暗号学的ハッシュ関数を新たな手法で置き換える必要性があると考察する．

本研究の貢献は TLS1.3 が普及した現在の環境において，従来の暗号化トラフィック分析手法が直面する可能性のある課題を実験的に提起した点である．今後の展望としては，TLS パラメタの揺らぎに対して頑強性のある新たな特徴量エンジニアリング手法の提案や，モデル構築によってこの課題を解決する必要がある．

参考文献

[1] Google LLC: ウェブ上での HTTPS 暗号化 - Google 透明性レポート (Thu Apr 24 2025), https://transparencyreport.google.com/https/overview?hl=ja.

[2] Sophos Ltd.: 約半数のマルウェアが通信の隠蔽に TLS を利用- Sophos News (Thu Apr 24 2025), https://tinyurl.com/sophos-tls-malware.

[3] Oh, C., Ha, J. and Roh, H.: A survey on TLS-encrypted malware network traffic analysis applicable to security operations centers, *Applied Sciences*, Vol. 12, No. 1, p. 155 (2021).

[4] Qualys, Inc.: Qualys SSL Labs - SSL Pulse (Thu Apr 24 2025), https://www.ssllabs.com/ssl-pulse/.

[5] Barradas, D., Novo, C., Portela, B., Romeiro, S. and Santos, N.: Extending C2 Traffic Detection Methodologies: From TLS 1.2 to TLS 1.3-enabled Malware, *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 181–196 (2024).

[6] Green, M., Droms, R., Housley, R., Turner, P. and Fenter, S.: Data Center use of Static Diffie-Hellman in TLS 1.3, Internet-Draft draft-green-tls-static-dh-in-tls13-01, Internet Engineering Task Force (2017).

[7] Gillmor, D. K.: TLS clients should reject static Diffie-Hellman, Internet-Draft draft-dkg-tls-reject-static-dh-00, Internet Engineering Task Force.

[8] Velan, P., Čermák, M., Čeleda, P. and Drašar, M.: A survey of methods for encrypted traffic classification and analysis, *International Journal of Network Management*, Vol. 25, No. 5, pp. 355–374 (2015).

[9] Papadogiannaki, E. and Ioannidis, S.: Acceleration of intrusion detection in encrypted network traffic using heterogeneous hardware, *Sensors*, Vol. 21, No. 4, p. 1140 (2021).

[10] Korczyński, M. and Duda, A.: Markov chain fingerprinting to classify encrypted traffic, *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 781–789 (online), DOI: 10.1109/INFOCOM.2014.6848005 (2014).

[11] Bartos, K., Sofka, M. and Franc, V.: Optimized invariant representation of network traffic for detecting unseen malware variants, *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, USA, USENIX Association, p. 807–822 (2016).

[12] Aitken, P., Claise, B. and Trammell, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, RFC 7011 (2013).

[13] McGrew, D. and Anderson, B.: Enhanced telemetry for encrypted threat analytics, *2016 IEEE 24th international conference on network protocols (ICNP)*, IEEE, pp. 1–6 (2016).

[14] John Althouse: Open Sourcing JA3 - Salesforce Engineering Blog, https://engineering.salesforce.com/open-sourcing-ja3-92c9e53c3c41/.

[15] 小林淳史，鈴木雅斗，佐々木満春：JA3 フィンガープリントによる DDoS 攻撃の検知及び防御手法の実践，情報通信システムセキュリティ研究会（ICSS），Vol. 124, No. ICSS-265, 電子情報通信学会，pp. 8–15 (2024).

[16] Cui, S., Han, X., Dong, C., Li, Y., Liu, S., Lu, Z. and Liu, Y.: MVDet: Encrypted malware traffic detection via multi-view analysis, *Journal of Computer Security*, Vol. 32, No. 6, pp. 533–555 (2024).

[17] Deri, L., Martinelli, M., Bujlow, T. and Cardigliano, A.: ndpi: Open-source high-speed deep packet inspection, *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, pp. 617–622 (2014).

[18] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y.: LightGBM: a highly efficient gradient boosting decision tree, *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Red Hook, NY, USA, Curran Associates Inc., p. 3149–3157 (2017).

[19] Zakir Durumeric: GitHub - zakird/crux-top-lists: Downloadable snapshots of the Chrome Top Million Websites pulled from public CrUX data in Google BigQuery., https://github.com/zakird/crux-top-lists?tab=readme-ov-file.

[20] Ruth, K., Fass, A., Azose, J., Pearson, M., Thomas, E., Sadowski, C. and Durumeric, Z.: A world wide view of browsing the world wide web, *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, New York, NY, USA, Association for Computing Machinery, p. 317–336 (online), DOI: 10.1145/3517745.3561418 (2022).

[21] Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3, RFC 8446 (2018).

[22] Wickramasinghe, N., Shaghaghi, A., Tsudik, G. and Jha, S.: SoK: Decoding the Enigma of Encrypted Network Traffic Classifiers, *2025 IEEE Symposium on Security and Privacy (SP)*, IEEE, pp. 1825–1843 (2025).

[23] Stratosphere Research Laboratory: Malware Capture Facility Project — Stratosphere IPS (Thu Jul 17 2025), https://www.stratosphereips.org/datasets-malware.