STIX2.0/TAXII2.0を用いたインディケータの自動収集と攻撃検知の自動化

藤本万里子[1]　松田亘[1]　満永 拓邦[1]

概要：ドメイン名やIPアドレスなどのC2サーバの情報は，標的型攻撃を検知するための有用な手がかりとなる．近年，情報共有が世界中で進んでおり，インディケータの取り扱いの自動化のために，脅威情報の標準記述形式であるSTIXや，STXIを交換するための仕様であるTAXIIの普及が進んでいる．共有された情報，いわゆるインディケータの活用局面として，過去に発生した攻撃を検知すること，および将来的に発生しうる攻撃に備えて利用することが挙げられる．STIXを活用した効果的な検知のためには，適切なタイミングでSTIX形式のインディケータを受信し，組織のログを突合する必要がある．本研究では，TAXIIを用いて，STIX 2.0形式のインディケータを自動的に収集し，オープンソースのログ分析エンジンであるElastic Stack上でプロキシログと突合することで，攻撃を効率的に検知する手法を提案する．

キーワード：標的型攻撃, STIX,TAXII,Elastic Stack, インディケータ

Automated indicators collection and attack detection using STIX2.0/TAXII2.0

Mariko Fujimoto[1]　Wataru Matsuda[1]　Takuho Mitsunaga[1]

Abstract: In detection of targeted attacks, indicators such as C2 server and IP address information can be useful. Information sharing scheme has been developed globally during the past years. A standardized format for describing cyber threat information called STIX, and transport mechanism for STIX called TAXII are getting popular to automate indicator handling. Shared information, in other words practical use of indicators serves two purposes: detecting malicious communication that occurred in the past and preparing for future attacks. For effective detection utilizing STIX, it is necessary to collect STIX format indicators in a timely manner and compare them with logs in the organization. In this research, we propose an effective method which collects STIX 2.0 indicators automatically with TAXII, compares indicators with proxy logs, and analyzes them on Elastic Stack, an open source log analysis engine.

Keywords: APT, STIX, TAXII, Elastic Stack, Indicator

1. はじめに

多数の組織が標的型攻撃の被害を受けている．標的型攻撃で使用されるマルウエアは, HTTP/HTTPSなどを使ってC2サーバと呼ばれる攻撃者のサーバと通信を行い, 命令を受信して感染を拡大させることが多い [1]．その際, 組織のプロキシサーバのログにC2サーバとの通信履歴が残る場合があり, それらを検知することで, 感染端末を特定できる場合がある．攻撃の検知を目的として, JPCERT/CCやIPAなどがハブとなって，C2サーバのドメイン名やIPアドレスなどの攻撃を検知するための情報をインディケータとして展開している．しかし, 発信元によって異なる記述形式であるため組織間での齟齬が発生したり, 検知の自動化が難しいという問題があり，脅威情報の記述を標準化するための仕様であるSTIX(Structured Threat Information

[1] 東京大学情報学環 セキュア情報化社会研究グループ
The University of Tokyo, Secure information society research group

eXpression)[2] と，STIX を組織間で自動的に交換するための標準仕様である TAXII(Trusted Automated eXchange of Indicator Information)[3] が策定されている．2017 年に公開された STIX 2.0[4]，TAXII 2.0[5] では，JSON[*1] などの汎用的に使われている技術が用いられることから，アプリケーションの実装やシステム間連携が容易になり，STIX/TAXII による情報連携の促進が期待される．ただし，STIX/TAXII を活用した効率的な攻撃検知を行うためには，適切なタイミングで脅威情報の情報収集と検知を自動化できるツールが必要になる．そこで，本研究では，TAXII 2.0 を用いて，STIX 2.0 形式のインディケータから C2 サーバの情報を自動的に収集し，オープンソースのログ分析エンジンである Elastic Stack[6] 上で組織のプロキシログと突合することで，攻撃を効率的に検知する手法を提案する．

1.1 標的型攻撃の概要

特定の組織の情報を窃取するなどの明確な目的を持った攻撃を標的型攻撃と呼ぶ．攻撃者は巧妙な手口を使って，目的を達成するまで執拗に組織内ネットワークに潜伏して攻撃を続ける．標的型攻撃で使用されるマルウエアは，C2 サーバと呼ばれる攻撃者のサーバと通信を行い，命令を受信し，組織内ネットワークで侵害範囲を拡大していき，最終的に機密情報窃取などの目的を果たす (図 1)．標的型攻撃では侵入自体を防ぐことは難しいが，攻撃に早期に気づき，対応することができれば，被害を最小限に抑えることが可能である．

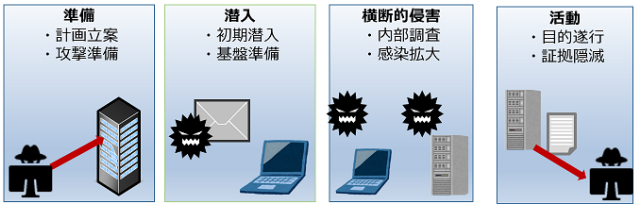

図 1　標的型攻撃のステップ

Fig. 1　Step of the targeted attack

1.2 インディケータを用いた攻撃の検知

組織内からインターネットに対する通信がプロキシサーバを経由している場合，感染端末と C2 サーバとの通信がプロキシサーバのログに残る場合がある．C2 サーバとの通信を早期に検知することで，感染端末を特定できる場合があり，被害を最小限に留めることができる．このような C2 サーバのドメイン名や IP アドレスなど，サイバー攻撃の活動を見つけるための情報のことをインディケータと呼ぶ．インディケータは攻撃を受けた組織から提供された情報に基づき，情報共有のハブ組織などを通じて，関連組織に通知される．インディケータを受信した組織は，インディケータに記録された情報を元に，自組織のプロキシサーバやファイアウォールのログを検索することで，該当する通信が自組織で発生していないかを確認し，感染の有無の確認などを行うことができる．

1.3 STIX/TAXII の概要

STIX(Structured Threat Information eXpression) は，2013 年に MITER によって策定された脅威情報の記述を標準化するための仕様である．TAXII(Trusted Automated eXchange of Indicator Information) は STIX を交換するための標準仕様であり，STIX で記述した脅威情報を TAXII によって交換することで，組織間で脅威情報の記述形式の統一化および情報連携の自動化が実現できる．活用例として，情報連携のハブ組織となる機関が TAXII サービス提供用のサーバを公開し，ISAC などの情報共有フレームワークに参加している各組織が，ハブ組織の TAXII サービスを仲介して脅威情報の交換を行う様なケースが考えられる (図 2)．図 2 の例では，被害組織である組織 A がハブ組織に STIX で記述されたインディケータ (以下，STIX インディケータ) を共有し，ハブ組織が被害組織の関連組織に STIX インディケータを配布する例である．標的型攻撃のインディケータは悪用されることを避けるために，通常配布する範囲が決まっている場合が多く，一連の STIX データの集まりは Data Collection と呼ばれる．

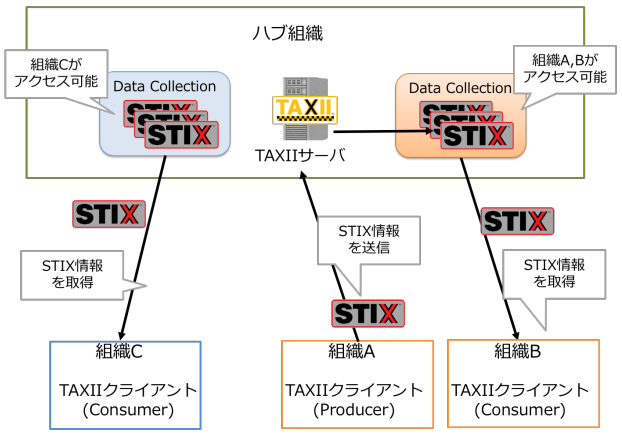

図 2　TAXII を用いた組織間での STIX 情報の交換の例

Fig. 2　An example on exchanging STIX among organizations using TAXII

2017 年には，STIX 2.0，TAXII 2.0 が公開され，記述形式が XML から JSON に変更された．TAXII では，STIX を HTTP/HTTPS で交換することが定められており，TAXII 2.0 では REST API[*2] を使用する．STIX 2.0，TAXII 2.0 では，アプリケーションの実装やシステム間連携が容易に

[*1] JavaScript Object Notation: JavaScript のオブジェクトの表記法をベースとするデータフォーマット

[*2] システム間連携のための Web ベースの API

なったことや, 2020 年の東京オリンピック対応などに向けて，STIX/TAXII による情報連携の促進が期待される.

2. STIX/TAXII を活用した攻撃検知における課題

2.1 自動化システムの必要性

STIX/TAXII を活用した攻撃検知における課題について述べる. STIX/TAXII はシステムによる自動連携をコンセプトとした仕様であり, STIX/TAXII を活用して効率的かつ効果的な検知を行うためには, 以下の作業を自動化できるシステムを持っていることが望ましい.

- TAXII による STIX の自動収集
- STIX のインポートやエクスポート
- STIX と組織のログの突き合わせ

そこで, 本研究では，無償で利用できるソフトウエアを使用して，これらの機能を実現するための手法について提案する.

2.2 検知のタイミング関する課題

前節で，STIX/TAXII の活用に際して適切な機能を持つツールが必要であることを述べたが，さらにツールの実装に関連する主な課題として, 検知のタイミング関する課題が挙げられる. インディケータを用いた攻撃検知においては, 攻撃を早期に検知でき，かつ検知の取りこぼしが少なくするためには，以下のタイミングでインディケータとログを突合する必要がある.

(1) インディケータ受信時：インディケータを受信した際は，インディケータに該当する通信が発生していないかを確認するために，受信したインディケータと過去のログを突合する必要がある．提供されるインディケータの数が増加した場合，インディケータ受信後，手作業の運用によって迅速にログを検索することは難しくなるため，システムによる自動化が重要となる．実際に，インディケータを受信していたにも関わらず，適切な処置をとっていなかったことが原因で，攻撃の発見が遅れ，情報漏洩の被害が発生した事例もある [11].

(2) インターネットへの通信の発生時：組織内のコンピュータからインターネットに対して通信が発生した際に，リアルタイムでプロキシログを収集し，それらの通信が過去に受け取ったインディケータの通信先に該当しないか，収集したログと過去のインディケータを突合する必要がある．この手法については，[8] にて紹介しており，本研究でも [8] で紹介している手法を利用する.

3. 関連研究

本章では、STIX/TAXII を利用した既存研究について述べる．Sandeep Narayanan らは，Wannacry などの攻撃を検知するための STIX の記述方法を提案してるが，それらを活用するシステムの実現方法については言及されていない [7]．Mihai-Gabriel IONITA らは，STIX/TAXII を用いて収集した脅威情報と，オープンソースの侵入検知システムである OSSEC のログを，SIEM[*3]である AlienVault に統合して分析することで，IoT デバイスなどの不審な挙動のリスクアセスメントを行うシステムを提案している [8]. 本研究は，DoS やパスワードクラックなどの不審な挙動をモニタリングする目的であり，インディケータから感染端末を検知している本研究とは目的が異なる．また，藤本らは STIX とプロキシログをログ分析ツールに集約し，リアルタイムにインディケータに該当する通信の発生を検知する手法を提案している [9](図 4)．この手法では，インターネットへの通信の発生時のリアルタイム検知は実現できるが，インディケータ受信時にリアルタイムに近いタイミングで検知を自動化することはできない.

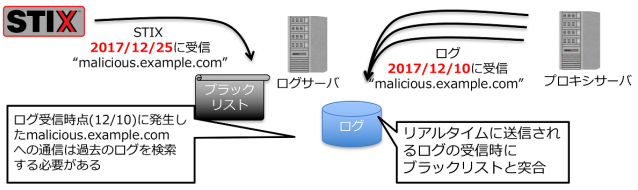

図 3 検知のタイミング関する課題

Fig. 3 Problems on the detection timing

4. 提案手法

4.1 提案システムの概要

2 章で述べた課題を解決するために, 本研究では, TAXII 2.0 を用いて STIX 2.0 形式のインディケータを自動的に収集し，ログサーバに集約することで, C2 サーバとの通信を効率的に検知する手法を提案する (図 5). 提案手法では, プロキシサーバのログと STIX インディケータを集約し，分析するためのソフトウエアとして，Elasticsearch 社より提供されているログ分析・可視化ツールである Elastic Stack を使用する.

提案手法では，[9] で提案されている手法に，TAXII サーバと連携することにより STIX インディケータを自動的に取得する仕組みを追加することで，インディケータ受信時においてもリアルタイムに近いタイミングで攻撃を検知する手法を提案する (図 6).

提案システムにおけるインディケータ受信時の検知の流れを以下に示す.

(1) TAXII サーバから STIX インディケータを受信する

(2) 受信した STIX インディケータから, 通信先の情報 (ド

[*3] Security Incident and Event Management の略で、セキュリティ製品やアプリケーションなどが出力するイベント情報を一元的に保管し、事象を把握するための技術

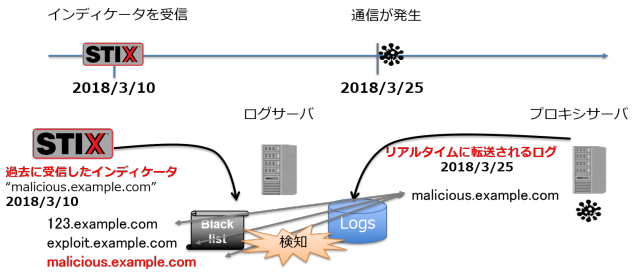

図 4　インターネットへの通信の発生時のリアルタイム検知 (既存手法)

Fig. 4　Real-time detection when communication to the Internet occurs

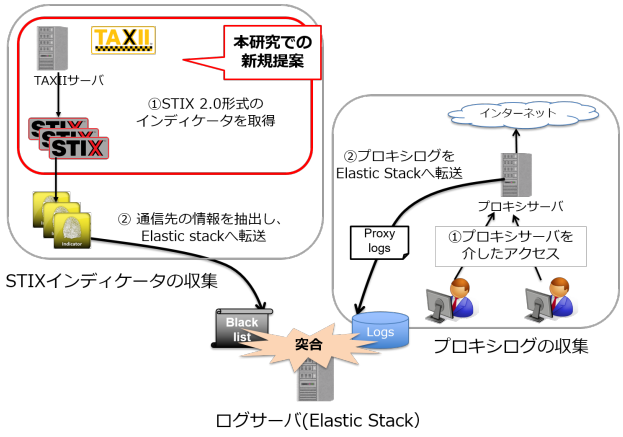

図 5　提案手法の概要

Fig. 5　Summary of the proposed method

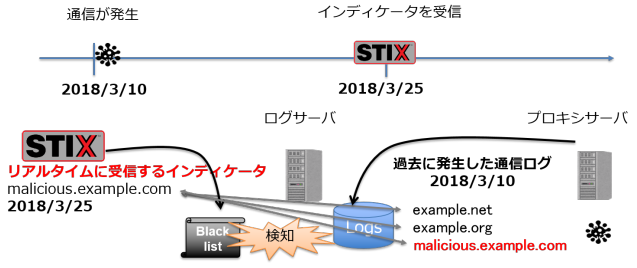

図 6　インディケータ受信時のリアルタイム検知 (提案手法)

Fig. 6　Real-time detection when receiving indicators

メイン名, IP アドレス, URL) を抽出し, それらに合致するデータがないか, Elastic Stack に蓄積されているプロキシサーバのログを検索する.

(3) 合致するログがあった場合, セキュリティ管理者にアラートメールを送信する. また、ログにインディケータに合致したことを示すフラグを付与して Elastic Stack へ転送する. 本フラグを付与することにより, インディケータに該当する通信ログの検索を Elastic Stack 上で行いやすくなる.

(4) STIX インディケータから抽出した通信先情報を Elastic Stack が保持するブラックリストに登録する.

STIX インディケータを自動的に取得し, ログとインディケータを突合することにより, インディケータの受信, ログの調査, 感染端末の特定を全て自動化することができる (図 7).

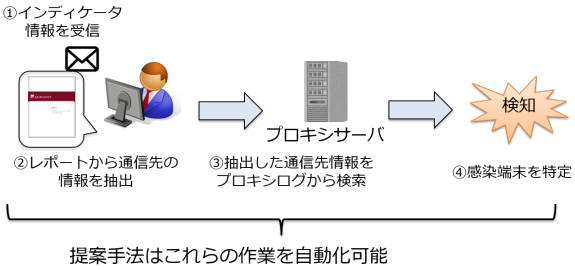

図 7　提案システムによるインシデントレスポンス時間の短縮

Fig. 7　Redution of the incident response time by the method

4.2　提案システムの構成

提案システムの構成を表 1 に, 構成を図 8 に示す. Elastic Stack は, 全文検索エンジンである Elasticsearch, 可視化ツールである Kibana, ログをパースし, 加工などを行うツールである Logstash, ログファイル転送エージェントである Filebeat などで構成される.

表 1　提案システムの構成

Table 1　Structure of the proposed method

	OS / ソフトウエア	バージョン
ログサーバ	OS	CentOS7.1.1503
	Elasticsearch	5.6.4
	Logstash	5.6.4
	Kibana	5.6.4
	cti-taxii-client	-
	Python	3.6
Proxy server	OS	CentOS7.1.1503
	Squid	3.3
	Filebeat[*4]	5.5.1
TAXII サーバ	cti-taxii-server	-
	medallion	0.1.0
	Python	2.7.5
	OS	CentOS7.1.1503

4.3　TAXII を用いた STIX 自動収集の詳細

本節では, STIX 情報の収集を自動化する仕組みの実装方式の詳細について述べる.

TAXII はサービスを提供する TAXII サーバとそれらのサービスを利用する TAXII クライアントで構成される.

4.3.1　TAXII サーバ

TAXII サーバは, STIX 情報を集中管理し、それらにアクセスするためのサービスを提供する Web サーバである. TAXII はクライアントサーバ型のシステムで, クライアントから TAXII サーバへリクエストを送り, サーバがレス

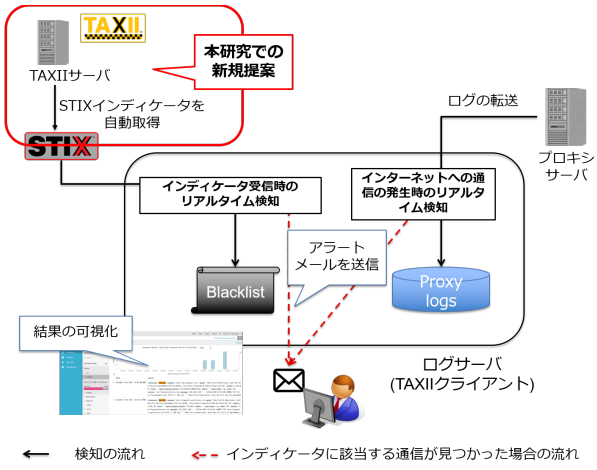

図 8　提案システムの構成

Fig. 8　Structure of the proposed method

ポンスを返すことで，TAXII サービスを利用することができる．表 2 は TAXII サーバが提供する代表的なサービスの例である．

TAXII サービスにおいて，クライアントとサーバでやり取りされるメッセージは HTTP/HTTPS 上で，REST API を使って行われる．図 9 は TAXII サービス (Get Objects サービス) でやり取りされるメッセージの例である．

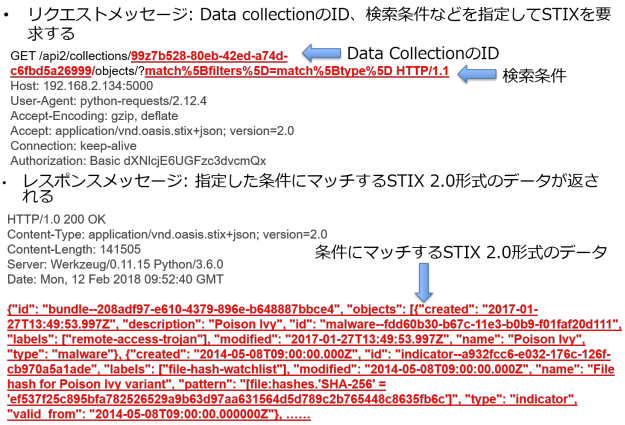

図 9　TAXII サービス (Get Objects) のメッセージの例

Fig. 9　An Example of TAXII service message(Get Objects)

TAXII サーバの機能を提供するための複数のソフトウエアが存在するが，本研究では TAXII 2.0 に対応しており，OASIS TC が公開しているオープンソースの TAXII 2 Server Library[12] を用いる．

4.3.2　TAXII クライアント

TAXII サーバを介して，TAXII サービスを利用し，STIX 情報を交換するエンドポイントのことを TAXII クライアントと呼ぶ．TAXII クライアントには，STIX を生成して送信する Producer と，STIX を要求する Consumer の役割が存在する．

提案手法における TAXII サービスの構成を，図 10 に示す．Producer の役割を持つ TAXII クライアントが STIX インディケータを TAXII サーバに送信すると，インディケータが TAXII サーバに格納される．ログサーバは TAXII クライアント (Consumer) となり，定期的に TAXII サーバから STIX インディケータを取得する．TAXII サーバに格納されている STIX 情報は，ログサーバから TAXII サーバに対して，Get Objects サービスを呼び出すことで取得できるが，STIX インディケータの取得および Elastic Stack への登録処理を自動化するために，ログサーバにスケジュール実行されるジョブプログラムを設置し，1 分おきに実行する．なお，TAXII クライアントの機能を利用するためには専用のライブラリが必要であり，本研究では，OASIS TC が公開しているオープンソースの TAXII 2 Client Library[13] を用いる．

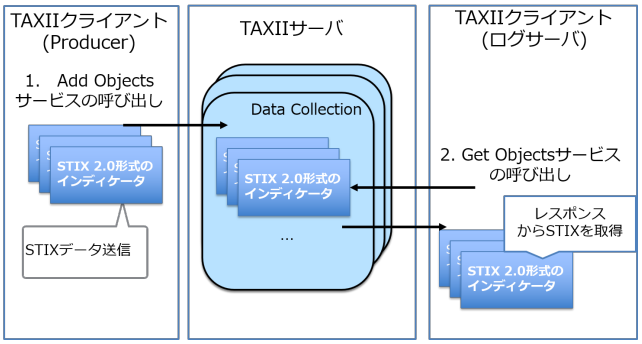

図 10　提案システム TAXII サービスの構成

Fig. 10　Structure of the TAXII service

5.　提案手法の検証

5.1　検証内容

提案手法を用いて, TAXII サーバから STIX 情報の自動収集を行うことで，インディケータの受信，ログの調査，感染端末の特定に要する時間が，手作業で調査を行った場合と比較して，どれだけ短縮されるかを検証する．インディケータ情報として，標的型攻撃のキャンペーンである APT1 のレポートを使用する．手作業についてはテキスト形式で記述されているレポート [14] を，提案システムについては，STIX 2.0 形式で記述されている [10] を使用する．手作業で表 3 に示す作業を実施し、作業にかかった時間と，提案システムによる検知時間を比較する．

5.2　検証結果

提案手法が検知に要する時間は，TAXII サーバに STIX インディケータが登録されるタイミングとジョブプログラムの起動タイミングに依存するが，STIX インディケータ登録後、ジョブプログラムの起動に要する時間は最大で 1 分要するため，最大にかかる時間を想定して評価を行なった．表 4，5 に示すとおり, 提案手法により，手作業に要する時間の約 9 割を削減することができた．

表 2　TAXII サービスの例
Table 2　Examples of TAXII service

サービス	概要	入力の例	応答の例
Discovery Resource	itaxii サービスを見つける	-	TAXII のバージョンなど
Collection Resource	Data Collection 情報を取得する	Collection の id	Data Collection の情報
Get Objects	taxii サーバから stix データを受信する	Collection の id，検索条件など	条件にマッチする stix データ
Add Objects	taxii サーバに stix データを送信する	Collection の id，STIX データ	結果コード

表 3　検証内容 (手作業)
Table 3　Evaluation(manual operation)

No	作業内容
1	インディケータ情報を受信
2	レポートに記載されているインディケータ情報から通信先の情報を抽出
3	抽出した通信先情報をプロキシログから検索

表 4　検証結果 (手作業)
Table 4　Evaluation result(manual operation)

No	作業内容	所要時間
1	インディケータ情報を受信	1 分
2	レポートに記載されているインディケータ情報から通信先の情報を抽出	5 分
3	抽出した通信先情報をプロキシログから検索	3 分 31 秒
計		9 分 31 秒

表 5　検証結果 (提案手法)
Table 5　Evaluation result(proposed method)

No	作業内容	所要時間
1	STIX インディケータ登録後、ジョブプログラムの起動に要する最大時間	1 分
2	ジョブプログラムが起動してから，アラートメールを受信するまでの時間	6 秒
計		1 分 6 秒

6. 終わりに

標的型攻撃の様に，迅速なインシデントレスポンスが要求される攻撃に対して, STIX および TAXII を用いて，インディケータ情報の取得と, インディケータとログの突合を自動化することで, 攻撃を早期に検知し, インシデントレスポンスにかかる時間を短縮できることが分かった．今後の課題として, 以下について検討を行う．

- ブラックリストやプロキシログの件数が増えると, 突合時の処理量が膨大になるため, インディケータの有効期限などを考慮した仕組みなどを考える必要がある．
- STIX は統一フォーマットであるものの, 一つのインディケータに複数の C2 サーバの情報を記述するなど, 様々な記述パターンが存在している．本研究では, 一つのインディケータに一つの通信先が記述されている場合は対応できるが, 複数の条件が記述されている場合は対応できない．今後は, インディケータの記述方法を更に調査し, より多くの記述パターンでも処理できるように改善を行う予定である．

参考文献

[1] LAC, 標的型攻撃対策指南書, p23, https://www.lac.co.jp/library/pdf/anti-apt_guidebook_ver1.pdf
[2] OASIS, Introduction to STIX https://oasis-open.github.io/cti-documentation/stix/intro
[3] OASIS, Introduction to TAXII https://oasis-open.github.io/cti-documentation/taxii/intro
[4] OASIS, STIX Version 2.0. Part 1: STIX Core Concepts, http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part1-stix-core.html
[5] OASIS, TAXII Version 2.0., https://taxiiproject.github.io/taxii2/
[6] Elasticsearch, The Open Source Elastic Stack https://www.elastic.co/products
[7] Sandeep Narayanan, Ashwinkumar Ganesan, Karuna Joshi, Tim Oates, Anupam Joshi and Tim Finin : Cognitive Techniques for Early Detection of Cybersecurity Events,
[8] Mihai-Gabriel IONITA, Victor-Valeriu PATRICIU : Secure Threat Information Exchange across the Internet of Things for Cyber Defense in a Fog Computing Environment，Informatica Economica vol. 20, no. 3(2016).
[9] 松田 亘, 藤本万里子, 満永 拓邦 : STIX 2.0 と Elasticserach を活用した攻撃検知手法の提案，暗号化とセキュリティシンポジウム (2018).
[10] MITRE, STIX 2.0 Threat Reports https://oasis-open.github.io/cti-documentation/examples/example_json/apt1.json
[11] 日本年金機構, 不正アクセスによる情報流出事案に関する調査結果報告書, https://www.nenkin.go.jp/info/index.files/kuUK4cuR6MEN2.pdf
[12] OASIS, OASIS TC Open Repository: TAXII 2 Server Library Written in Python, https://github.com/oasis-open/cti-taxii-server
[13] OASIS, OASIS TC Open Repository: TAXII 2 Client Library Written in Python, https://github.com/oasis-open/cti-taxii-client
[14] MANDIANT, APT1