標的型メール攻撃に対する知的ネットワークフォレンジックのための予兆検知と対策方法の提案

比留間裕幸[†1]　橋本一紀[†2]　柿崎 淑郎[†3]　八槇　博史[†4]
上原哲太郎[†5]　佳山 こうせつ[†6]　松本 隆[†7]　佐々木良一[†8]

近年，標的型メール攻撃をはじめとするサイバー攻撃が増加の一途を辿っているが，入口対策の限界やセキュリティ技術者の不足，組織内で収集されるログの不整合等の問題により一般的な組織では適切な対応が難しい現状がある．よって著者らは，運用時には収集するべきログの管理や攻撃事象の特徴から検知，分析，対応を行い，ネットワーク整備段階では計画支援機能を備えた LIFT(Live and Intelligent Network Forensics Technologies)システムの開発に着手した．LIFT の検知機能は攻撃事象の粒度を階層構造で 7 つに分け，ボトムアップ式に確信度を上げることで事象の推定を行う．確信度をあげる上で関連する別の兆候の調査や通常では収集されないログの収集，マルウェア等の兆候誘発等を行う．また，推定される攻撃事象や兆候を否定する反証の調査も行うことで一方向でない広い視野での検知が可能となる．応急対応機能は推定された事象に対して，有効な応急対応策を算出し，自動的な対策案実施や管理者への指示を行う．この提案方式を実現する上で，ルールベースでの記述が可能となる JBoss Drools を用い，開発を行った．その結果，ルールを発火させることでソースから兆候検知が可能となることがわかった．

Proposal of warning detection and protection measures for Intellectual network forensics against the targeted email attacks

HIROYUKI HIRUMA[†1]　KAZUKI HASIMOTO[†2]　YOSHIO KAKIZAKI[†3]
HIROHUMI YAMAKI[†4]　TETSUTARO UEHARA[†5]
KOSETSU KAYAMA[†6]　TAKASHI MATSUMOTO[†7]　RYOICHI SASAKI[†8]

Graduate School of Information Media,　Tokyo Denki University
5,　Senju-Asahi-cho,　Adachi-ku,　Tokyo,　120-8551 JAPAN

[†1]hiruma@isl.im.dendai.ac.jp*,　[†2]hashimoto@isl.im.dendai.ac.jp
[†3]kakizaki@im.dendai.ac.jp,　[†5]yamakih@mail.dendai.ac.jp
[†8]sasaki@im.dendai.ac.jp

1. はじめに

最近，サイバー攻撃は厳しさを増しており，それに伴ってネットワークフォレンジックの必要性が増している[1]. 特定の個人や組織を，不正メールをトリガーとして集中的に攻撃する標的型メール攻撃が増加してきている．企業などの組織では怪しいメールを開かなくするための教育がおこなわれているが，巧妙な標的型メール攻撃に対しては，このような入口対策の有効性は低い[2]. また内部の攻撃対策においても，分析に必要なログが収集されていない，ログに出力されている時間に不整合がある等，攻撃対策が適切に行えている企業は少ない[3].

近年このような攻撃に対応するため，SIEM（Security Information and Event Management : セキュリティ情報およびイベント管理）システムが注目を浴びている[4]. SIEMシステムは，①ログ管理統合ツールに，②セキュリティ脅威に対するリアルタイムな検知機能を追加し総合的に対応できるようにしたシステムと言っていいだろう.

①のログ管理を適切に行うためには，ネットワーク系のログなどの収集・分析・保存を，証拠性を確保しつつ実施するためのネットワークフォレンジック技術が不可欠であり，②の機能を追加したものは，ライブ・ネットワークフォレンジックシステムということも可能であろう．このライブ･ネットワークフォレンジックシステムともいうべき，SIEMシステムは，あまりにも運用者の能力に頼りすぎているという面がある．SIEMシステムを使いこなせるのは，非常に高い能力を持つ複数の運用者がいる組織だけであって，一般の組織では適切な対応が不可能である．さらに高い能力を持つセキュリティ技術者が全体を通して不足している現状もある[5]. このため，運用者へのガイド機能や，半自動運転機能を導入した知的システムにし，高い運用能力を持つ運用者がいない組織であっても対応できるようにしていく必要があると考えた．すなわち，AI (Artificial

†1, †2, †3, †4, †6, †8 東京電機大学
Tokyo Denki University
†5 立命館大学　教授
Ritsumeikan　University
†7 ネットエージェント株式会社
Net Agent Company

Intelligence)を利用したLIFT（Live and Intelligent Network Forensics Technologies）システムを開発する必要があると考えた[6].

このため著者らは，2013年度後期に東京電機大学のサイバー・セキュリティ研究所内にLIFTプロジェクトを設置し，他大学や企業の専門家の協力も得て，LIFTシステムの開発に着手した．本システムは収集するべきログの管理や攻撃事象の特徴から検知，分析を行い，応急対応を行う機能およびインシデント後の組織内ネットワークにおける計画支援機能を持つ.

LIFTシステムと同様な機能を持つシステムは，いろいろなところで研究を実施している可能性は高いが，正式に発表されたものはない.

本稿では，第1章で背景，2章で本稿にて使用する用語の定義，3章でLIFTシステムの概要，4章で提案方式，5章で開発概要を述べる.

2. 用語定義

本論文において用いる，各用語の定義を以下に示す.

表 1. 各用語の定義

名称	説明
ケース	過去に起こった出来事/事件
フェーズ	発生した出来事における現在の段階
事象	解決するべき案件や課題のこと
兆候	物事が起こった時に 変化する重要な状態変移，その特徴
ソース	兆候を発行する発行元情報 ログや IDS のアラート
LIVE メモリ	端末等の揮発性情報
WM	ワーキングメモリ．兆候の保存を行う
原因プロセス	通信の際にコンピュータ上で実際に動作を行ったプロセス
インシデント レスポンス	インシデント発生時の緊急対応のこと 応急対応と同義で用いる
事後対応	インデント終息時に再発を防止するために行う対応のこと

3. 標的型メール攻撃とは

標的型メール攻撃とは標的組織への攻撃や秘密情報の窃取を狙ったサイバー攻撃の一種である．一般的に行われる手法として，メールの送信者名や添付ファイル名の偽装やメール本文を標的が関心を抱きそうな内容にすることで，不正プログラムを巧みに実行させる手法が挙げられる．侵入に成功した攻撃者は標的内部で目的の情報を探索し，情報窃取を行う．標的型メール攻撃の攻撃シナリオは IPA によって図 1 のように定義されている[7]．　Ⅰの侵入段階での検知とブロックが最も効果的であるが，利用されるマルウェアの特性や攻撃者による巧妙な攻撃により困難であるため，Ⅱ，Ⅲ，Ⅳフェーズでの総合的な兆候検知が現実的である[8]

図 1. 攻撃のフェーズ分け

4. LIFT プロジェクト

LIFTプロジェクトの全体概要を図2に示し，LIFTプロジェクで開発中の主な機能と役割を以下に示す.

- 運用段階(LIFTシステム)

① 自動運転での攻撃事象推定機能（5.2節参照）

② 対策方法提案機能（5.3節参照）

- 実験および知識獲得機能

③ 状況認識知識獲得用実験機能

LIFTシステムにおいて，攻撃事象を再現する実験により，検知やログ分析に用いる知識を獲得し，知識の精度を高めていくことでLIFTシステムの検知機能や推定機能を改善していく.

④ 計画支援機能

計画支援は同プロジェクトにおいて，イベントツリーを用いて最適な対策案を算出する提案がなされている．詳しくは参考文献を参照[9].

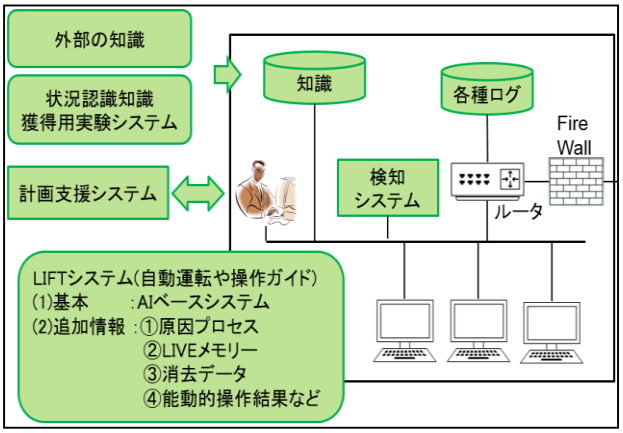

図2. LIFTプロジェクトの全体概要

LIFTプロジェクトにおいて開発中の運用段階のシステムであるLIFTシステムでは標的型メール攻撃の攻撃ケースや攻撃フェーズの早期検知と分析，証拠保全の半自動化とネットワークログ取得，管理のインテリジェント化を行うことを目的としている．基本は衆議院や大手重工に対する攻撃をベースとするが，それらの変形も対象とする．LIFTシステムの①，②機能をまとめた全体フローを図3に示す．

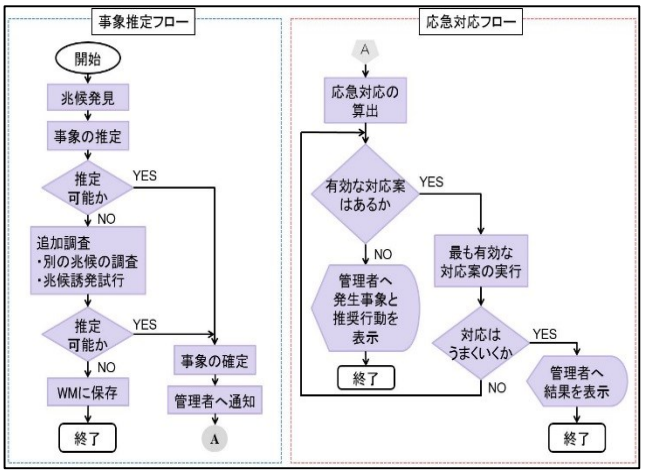

図3. LIFTシステムの全体フロー

5. 提案方式

本稿の提案方式では前章で述べたLIFTシステムを実現するものである．5.1節ではLIFTシステムにおける攻撃事象等の各種名称を説明し，5.2，5.3節では前章①，②の手法を提案するものである．

5.1. 各種名称の階層構造

LIFTシステムにおける各種名称の階層構造を図4および以下に箇条書きにて示す．

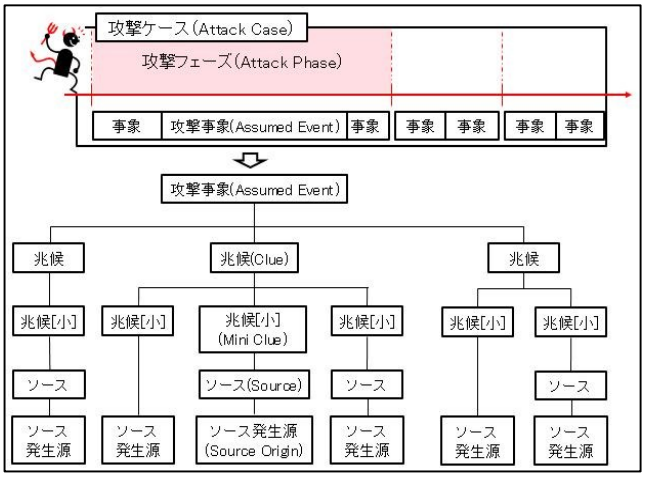

図4. 攻撃における各種名称の階層構造

(1) 攻撃ケース(Attack case)

攻撃ケースは今現在晒されている攻撃が，過去起こったどのような攻撃に似ているのかを表したものである．過去の攻撃と比較することで，受けた被害や賠償金，使われた進入路や脆弱箇所，行われた対策などを明確に表すことが可能な為，インシデントレスポンスの際に素早い対応が可能となる．攻撃ケースが表す粒度は衆議院型標的型攻撃や大手重工業型標的型攻撃のような形で表す．

(2) 攻撃フェーズ(Attack phase)

攻撃フェーズは今現在晒されている攻撃または検知した攻撃事象が，概要にて示したIPAの攻撃シナリオのどのフェーズに当たるのかを表したものである．攻撃フェーズから攻撃者の侵攻度や自組織の現状を推定することで，被害を最小限に抑える対策の立案，実施や証拠保全作業といったリアルタイムな対応が可能となる．攻撃フェーズが表す粒度は図1のⅠからⅣに示す通りである．

(3) 攻撃事象(Assumed Event)

攻撃事象は今現在晒されている恐れのある攻撃において，攻撃者や攻撃者が使用するツール，マルウェアが行う攻撃試行を表したものである．推定された攻撃事象から，推定される攻撃フェーズにおける現状や攻撃者の行動，使用ツールの予測が可能となり，実業務に影響を及ぼさない小規模または効果的な対応の立案，実施や攻撃者の攻撃パターンの予測が可能となる．攻撃事象が表す粒度は「C&Cサーバとの通信」や「内部のシステム情報を探索する」といった形で表す．

(4) 攻撃事象における兆候(Clue)

攻撃事象における兆候は攻撃事象に対して粒度を細分化したものである．攻撃事象と兆候は１対多の関連を持ち，多数ある兆候それぞれに，事象における確信度を付加することで攻撃事象への確信度を上げる役割を持つ．兆候が表す粒度は「内部のシステム情報を探索する」に対して「SMB，ファイル共有の探索」や「クライアント情報(netコマンド，プリフェッチド，システム情報)等の収集」等である．

(5) 攻撃事象における兆候（小）(Mini Clue)

攻撃事象における兆候（小）は攻撃者が行った行動(攻撃事象)によって各ネットワーク機器や端末に現れる異変や異常であり，前述した兆候の粒度を細分化したものである．兆候と兆候（小）は１対多の関連を持ち，事象における兆候と同様に兆候（小）はそれぞれに確信度が付加されており，兆候への確信度を上げる役割を持つ．兆候（小）が表す粒度は「SMB，ファイル共有の探索」に対して，「ネットワークセンサーに135/tcp検知」や「Dst端末やSrc端末で135/tcp検知」である．

(6)　ソース(Source)

ソースはソース発生源によって出力されたログのことである．本稿の提案方式において，兆候（小）観測ルールプログラムに則り，兆候検知を観測する為に利用するものである．ソースの例として以下の図5に示す．下図はプロキシのログにおいて，ポート443/tcp以外の通信を，CONNECTメソッドを用いてトンネリング要求しているという兆候（小）を検知した時の図である．

```
192.168.100.100 - -[13/Jan/2014:15:19:50 +0900]
"CONNECT www.korchambiz.net:8810 HTTP/1.1" 403
 3685 "-" "Mozilla/5.0
(Windows NT 6.3; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/31.0.1650.63
Safari/537.36" TCP_DINIED:NONE
```

図5. ソースの一例

(7)　兆候（小）が観測されるソースの発生源(Source Origin)

ソース発生源は兆候（小）を発行する各発行元のネットワーク機器や端末，検知ツールである．ソース発生源の例としてプロキシや重要サーバ，IDSやDst，Src端末のログや同プロジェクトで提案と開発がなされた原因プロセス情報を利用した不正通信原因推定ツールなどがあげられる．このツールは不正通信を実施したプロセスの特定と挙動の把握を行うことが可能なツールであり，このツールにより，既存のツールでは収集の難しかった，どのコンピュータ上で実際にプロセスが動作を行ったかを把握することが可能である [10].

5.2.　自動運転での攻撃事象推定機能

5.2.1.　概要

LIFTの事象推定機能では図4にて述べた攻撃の各階層構造をボトムアップ式に調査することで，兆候から事象推定を行う．事象からどのような兆候が発生し，どのようなソースに異常が現れるかをルール記述しておき，攻撃事象発生時には，逆演算することにより事象の推定を行う．事象と兆候の関係をテーブルに表したものを図5に示す．

フェーズ	事象＼兆候	不自然なプロセスの立ち上がり	プロキシ：プロキシを経由しない通信	プロキシ：プロキシ認証試行に規則性	プロキシ：443以外のCONNECTメソッドを利用した通信	業務に不要なソフトのインストール	ファイル共有試行	サーバへの不自然な時間の認証	業務外通信
基盤構築フェーズ	端末が不正プログラムを起動								
	C&Cサーバへ接続								
	ユーザ端末のuser権限奪取								
	感染端末のシステム情報窃取								

図6. 兆候事象発見テーブル

5.2.2.　事象推定フロー

攻撃事象推定機能のフローチャートを図7に示す．事象推定では図4にて示した兆候（小）から導かれる兆候の発見を検知のスタートとする．事象の推定が可能になるまで，WM(ワーキングメモリ)やその事象に関連する根っこの別の兆候の追加調査を行う．追加調査を行う上で，兆候における兆候（小）を観測する各種機器，端末のログ収集が業務ルールやログファイル肥大化を避ける為に取られていない場合や収集されていない場合がある．その場合には機器が自動的にログ収集を行い，事象推定の確信度をあげていく．関連するソースが収集されている場合には，意図的な通信の切断などといった兆候誘発を行うことで，通常業務では起こりえないマルウェアの行動や攻撃者による内部ネットワーク調査といった不審な足跡を洗い出す．

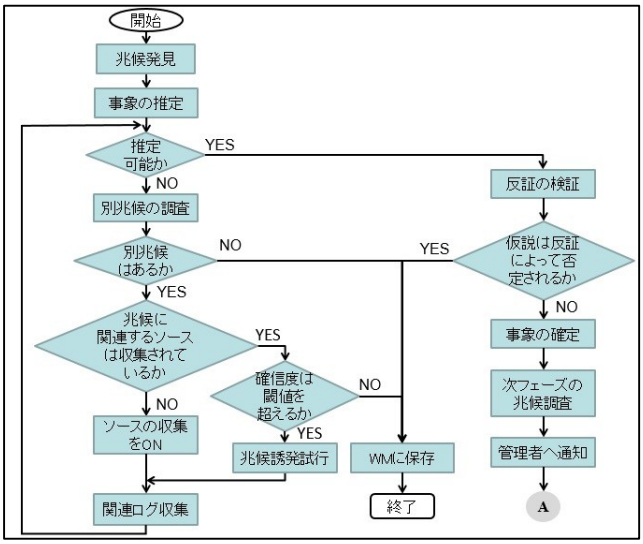

図7. 提案方式の事象推定フロー

兆候の調査を繰り返し，攻撃事象の推定が可能になった段階で,推定される仮説を否定する反証の調査を開始する．仮説における反証の例を表2に示す.反証とは推定された事象，兆候に対してそれぞれマイナスの確信度を持ち，それらに対して懐疑を行うことで，より正確な検知予測が可能となる．反証によって，推定された事象が否定された場合には観測された兆候をWMに保存し，事象推定フローを終了する．推定された事象が否定されなかった場合，攻撃事象の確定を行い，その攻撃事象の次フェーズや関連する攻撃事象の調査をインシデントレスポンスと同時並行で行う．また，推定された攻撃事象が反証の調査を行う僅かな間にも進行してしまう場合がある．よって，重要度の高いフェーズや攻撃事象，兆候の場合や推定されている攻撃事象の確信度が高い場合には反証の調査を行いながら，インシデントレスポンスのフローへと移行することが求められる．

表2．反証の一例

兆候	仮説	反証
ブラウザ履歴にブラックリストURLへのアクセス履歴が存在する	該当ブラウザで対象URLへアセスした	該当ページのキャッシュ情報や文字列痕跡が存在しない
		対象webサーバの同刻のログにレコードは存在しない
	該当ブラウザでURLを入力した	IEの場合，レジストリ内のTypedURLにはレコードが存在しない
		ブラウザの関連データ以外に該当URLの文字列，痕跡は存在しない
	該当ブラウザでURLのクリックをした	過去の履歴内に該当するURLへアクセスした履歴やキャッシュは存在しない
ブラウザキャッシュにブラックリストURLのデータ存在	該当ブラウザでURLへアクセスし，ページを表示した	キャッシュはURLへアクセスした時点のデータ内容ではない
		キャッシュは人間が扱うデータではない

5.3.　対策方法提案機能

5.3.1. 概要

前節にて推定された攻撃事象に対して，その事象を止める上で有効なインシデントレスポンスを自動で実行する．攻撃事象とインシデントレスポンスの対応をテーブルに表したものを図7に示す．このテーブルは攻撃事象と有効なインシデントレスポンスの対応を表したものである．

フェーズ	事象＼対応	該当端末のルータで遮断	該当ポートのルータで遮断	該当通信のドメインのルータで遮断	該当端末のインバウンド通信の遮断	該当端末のアウトバウンド通信の遮断	該当ネットワーク遮断	該当端末が所属するネットワークの隔離	ネットワーク全体の遮断	該当端末の隔離
基盤構築フェーズ	端末が不正プログラムを実行	有効でない	有効でない	有効でない	有効でない	有効でない	有効でない	有効でない	有効でない	有効
	C&Cサーバへ接続	有効	有効	有効	有効	有効	有効	有効	有効	有効
	ユーザ端末のuser権限奪取	有効でない	有効でない	有効でない	有効	有効	有効	有効でない	有効でない	有効
	感染端末のシステム情報窃取	有効でない	有効でない	有効でない	有効でない	有効でない	有効でない	有効でない	有効でない	有効

図8. インシデントレスポンステーブル

5.3.2. 応急対応フロー

応急対応フローを図9に示す．提案方式の応急対応では事象推定フローにて確定された攻撃事象に対して有効な対策案を算出し実行する．算出方法は確定した事象や観測した兆候を封じ込める上で適している，また付随する攻撃者の行動を抑止するもの，通常業務に支障が出ない点を元に選択する．もし対策案が算出出来ない場合には，重大なインシデントとして通知し，即時ネットワーク遮断やセキュリティ企業への連絡といった推奨すべき行動を提案する．

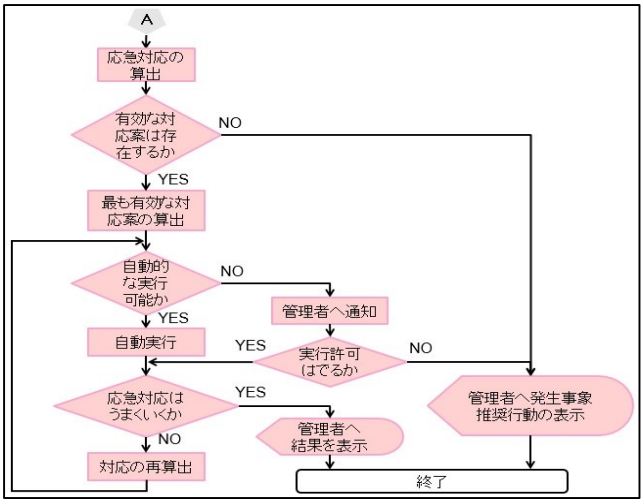

図9. 提案方式の応急フロー

6.　システム開発のための予備実験

6.1 JBoss Drools とは

提案方式を実現する上で，JAVAなどの手続き型言語を用いる方法もあるが，ルールプログラムとアプリケーションプログラムを分離した設計が可能な点やJavaで書かれたオープンソースエンジンである点からJBOSS Droolsを用いてみることとした．

兆候やソース等の関係を節3.2.1にて述べた階層構造別にルール化し，そのルールを発火させることで節4.2や4.3で述べたフローを表現する．ルールベースシステムは一般的に以下の図10のような構成となっている．プログラムは実行をつかさどる推論エンジンと「何を(what)」行うかを記述するルール部分とに別れており，推論エンジンはルールエンジンの「どのように」の実行部分の順序を一括で管理している[11]．

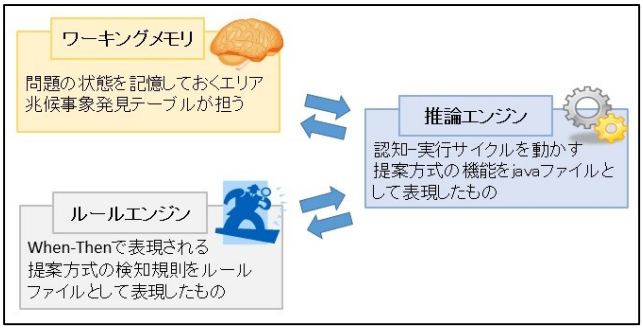

図10. ルールベースシステムの構成

6.2 プログラム概要

予備実験のための擬似的なプログラム開発は以下のように行っている．ルールファイルの一部を図11に示す．下図は関連ソースが収集されている場合をルールとして記述しており，Whenにてソースルールに定義されているソースと兆候，事象の関連をチェックし，Thenにて確信度の計算と出力処理を記述している．agenda-groupとは関数のような役割を果たし，ルールをまとめるものである．agenda-groupを利用して発火させたいルール群を呼び出すことが出来る．

```
rule "Support"↓
        agenda-group "Aggregation"↓
    when↓
      sign : Sign ()      event : Event ()↓
      support : Support (signId == sign.getID() && eventId == event.getID() &&
effective != true)↓
      Detected (signId == sign.getID())↓
      sourceSupport : SourceSupport(signId == sign.getID() && collectFlag == tru
e)↓
    then↓
      support.setEffective(true);↓
      update (support);↓
      System.out.println ("※Rule Support fired.");↓
      System.out.println ("Assumed Event" + event.getID() + "(" + event.getDescr
iption () + ") is supported by detected Clue " + sign.getID() + "(" ↓
      + sign.getDescription() +  ") with score(確信度) = " + support.getScore();
↓
end↓
```

図11. ルールベースシステムの構成

この開発プログラムはソースによる兆候の検知と確信度の管理，該当ソースが収集されていない場合のソース収集ON機能を擬似的に表現したものである．

開発したプログラムを実行すると以下のようになる実行画面を図12に示す．今回のプログラムにおいて「プロキシを経由しない通信」の兆候が検知されたと設定した．今回のプログラムの流れを以下に箇条書きにて示す．

1. 「プロキシを経由しない通信」の兆候をソースより検知した(1行目)
2. ルールプログラムが発火し，検知した兆候に関連した事象「C&Cサーバへの通信」の発生が推定された．(2行目)
3. 検知した兆候だけでは事象を確定出来なかった為，関連する別の兆候のソースを調査(3行目中の処理)
4. 別の兆候のソースを調査する上で該当ソースが収集されていなかった為，ソース取得をONにした(7行目)
5. ONにした結果，事象の確信度を計算し，出力する．(10行目～12行目)
6. 最も確信度が高い事象を表示(16行目)

```
 1 Clue:308(プロキシを経由しない通信)がDetectされました↓
 2 Assumed Event:305(C&Cサーバへの通信)の可能性があります↓
 3 ↓
 4 ※Rule NotSupport fired.↓
 5 Mini Clue: 308(プロキシを経由しない通信) のソース 201(Router_log)は取られていま
 6 せん↓
 7 Mini Clue: 308 のソース 201の取得をONにします↓
 8 ↓
 9 ※Rule Support fired.↓
10 Assumed Event305(C&Cサーバへの通信) is supported by detected Clue 308(プロキシを
11 経由しない通信) with score(確信度) = 5↓
12 Certainty for Assumed Event305  is 5↓
13 ↓
14 ※Rule Calculate Certainty fired.↓
15 =====↓
16 Most likely event is Assumed Event(C&Cサーバへの通信) with certainty(確信度) = 5↓
17 =====↓
```

図12. 開発プログラム実行画面

6.3. 考察

予備実験の為のプログラム開発や予備実験を通じて，事象推定の為の兆候検知ルール等の記述が可能であることを確認した．また，ルールプログラムや推論エンジン等も分離した設計になっているため，変更が容易に行えることがわかった．

今回の擬似的な実装は兆候から事象を判定する簡単なプログラムであった為，処理時間は非常に短いものであったが，提案方式や攻撃の階層構造をルールにて表現する上で，プログラムの最適化を検討していく必要がある．

7. おわりに

本稿では LIFT システムの基本構想や概略仕様を示すとともに，自動運転での攻撃検知と攻撃事象推定機能，および対策方法の算出実行機能の提案を行った．併せて簡単な例題に対し，ルールベースシステムである JBOSS Drools を用いプログラム開発を行い，ルール発火をさせることでソースから兆候検知が可能であるという知見を得た．

今後，検知における兆候のトリガーの設定や検知機能のさらなるプログラム開発，応急対応時の管理者へのわかりやすいガイド機能が行えるインターフェースの開発，揮発情報を保全するメモリフォレンジック機能の導入，また評価実験を行っていく予定である．

謝辞

本研究に際して，様々なご指導を頂きましたLIFTプロジェクトの関係者に深謝いたします．また，日常の議論を通じて多くの知識や示唆を頂いた情報セキュリティ研究室の皆様に感謝します．

参考文献

1) IDF 特定非営利活動法人デジタル・フォレンジック研究会
ネットワーク・フォレンジックの経験から得られたこと
http://www.digitalforensic.jp/expanel/diarypro/diary.cgi?no=410&continue=on

2) IPA 独立行政法人情報処理推進機構
「新しいタイプの攻撃」の対策に向けた設計・運用ガイド
http://www.ipa.go.jp/files/000017308.pdf

3) @IT atmarkkIT
いま，企業が情報を守るために必要な防御策とは
http://www.atmarkit.co.jp/ait/articles/1307/30/news004.html

4) 日立ソリューションズ，“SIEMとは”
http://securityblog.jp/words/714.html

5) NISC 内閣官房セキュリティセンター
サイバーセキュリティ戦略［案］
http://www.nisc.go.jp/active/kihon/pdf/cyber-security-senryaku.pdf

6) 佐々木良一，上原哲太郎，松本隆
標的型攻撃に対するネットワークフォレンジック対策の現状と今後の展望
Computer Security Symposium 2013

7) IPA 独立行政法人情報処理推進機構
標的型メール攻撃対策に向けたシステム設計ガイド
http://www.ipa.go.jp/security/vuln/newattack.html

8) APT対策入門
特定非営利活動法人 日本セキュリティ監査協会
APTによる攻撃対策と情報セキュリティ監査研究会

9) 橋本 一紀, 上原 哲太郎, 松本 隆, 佳山 こうせつ 柿崎 淑郎, 佐々木 良一
標的型メール攻撃に対する計画･運用問題解決のためのイベントツリーを用いた最適な対策案の選定手法の提案
DICOMO2014シンポジウム

10) 三村聡志， 佐々木良一
プロセス情報と関連づけたパケットを利用した不正通信原因推定手法の提案
DICOMO2014シンポジウム

11) ビジネスルールの館
プロダクションシステム(ルールベースシステム)とは
http://www.iluminado.jp/businessruleruleengine/rule-base-etc/17-production-system-rule-base.html