一般社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

新たに発生した異常に対応する説明を提示するシステム

大下 裕一† 村田 正幸†

†大阪大学 大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5
E-mail: †{y-ohsita, murata}@ist.osaka-u.ac.jp

あらまし インシデントが発生した際には、管理者はそのインシデントに対して、迅速に対応することが求められる。インシデントの検知のみならず、発生したインシデントに関する情報を管理者に提示することが、インシデントへの迅速な対応を行うために有用である。インシデントに関する情報を提示する方法として、これまでに対応したインシデントに関する情報を蓄え、学習することにより、新たに発生したインシデントと類似した過去のインシデントに関する情報を提示するという方法が考えられる。しかしながら、この方法では、新たに発生したインシデントが過去に発生したいずれのインシデントとも異なる場合であっても、誤って、過去のインシデントに関する情報を提示してしまい、インシデントの初動対応をかえって遅らせてしまうことも考えられる。

そこで、本稿では、新種のインシデントに対しても、管理者に誤解させることなく、当該インシデント対応のためのヒントとなる情報を提示する方法について議論する。本稿では、新たなインシデントが発生した際には、過去の類似した事象の名前を提示するとともに、類似事象との相違点がある場合には、相違点についても提示する。本稿では、本システムを実装し、正しく情報提示できることを示す。

キーワード 機械学習, インシデント, 説明

Explanation of newly detected anomalies

Yuichi OHSITA† and Masayuki MURATA†

†Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871 Japan
E-mail: †{y-ohsita, murata}@ist.osaka-u.ac.jp

Abstract When an incident occurs, the administrator is required to handle the incident. It is useful not only to detect an incident but also to present information about the incident. One approach to presenting information on incidents is to learn the explanation of incidents from the past incidents. However, this method may present wrong explanation if the newly detected incidents are different from any past incidents.

In this paper, we discuss a method for presenting information on the detected incidents without misleading administrators even if the incidents are different from any past incidents. In this paper, we develop a system that presents the name of the incidents similar to the detected incidents and the difference from the incidents.

Keywords Machine learning, Incident, Explanation

1. はじめに

従来からインターネットに接続されきたパソコンやスマートフォンに加え、IoT 機器と呼ばれる様々な機器がインターネットに接続されるようになった。インターネットに接続される機器が増加するにつれ、それらの機器が攻撃者から狙われるなど、セキュリティリスクが高くなっている。すでに、IoT 機器を対象とした攻撃も実際に発生している。2016 年には、Mirai と呼ばれる IoT 機器を対象としたマルウェアが登場し、その亜種の感染も広がっている[1]。今後、IoT 機器を対象とした新たな種類の攻撃が発生する可能性もあり、IoT によるサービスを提供する業者は、それらの新たな攻撃に対しても、迅速な対処が求められる。

攻撃に対して迅速な対応を行うためには、異常検知技術[2]を用い、インシデントを迅速に検出するだけでなく、そのインシデントについて分析することが必要となる。インシデントを分析することにより、外部からポートスキャンを受けた場合や、ブルートフォース攻撃を受けたものの、侵入に失敗した場合のように、被害が発生しておらず、早急の対処は必要ではない場

合なのか、あるいは、機器への侵入が成功し、機器が踏み台として用いられている場合のように、迅速に対応することが必要な場合なのかを把握することが可能となる。

また、インシデントの分析の結果、過去に同種のインシデントが発生し、対処した経験があり、対処方法の情報の蓄積があれば、その情報をもとに現在発生しているインシデントの対応方法を得ることも可能となり、迅速なインシデント対応が可能となる。

我々は、これまでに、過去に発生したインシデント情報を蓄積し、新たにインシデントが発生した際に、過去の類似インシデントを検索し、その情報を提示するシステムを構築した[3]。このシステムでは、インシデントの特徴(例えば､異常検知されたフローの情報)、インシデントの対応結果として、インシデント名やインシデントの対応方法の情報を蓄積し、新たにインシデントが発生した際には、当該インシデントの特徴をキーとして、蓄積されたインシデント情報を検索し、類似したインシデントに関する情報を提示する。その際、単に特徴量が類似したインシデントの情報を検索するのではなく、同一種類のインシデントの情報を抽出することができるように、機械学習を用い、同種のインシデントの特徴量が近い位置に写像されるように学習した写像関数を通し、写像関数の出力が近いインシデントを取得する。このシステムにより、システムに投入されたインシデントの数が少ない種類のインシデントがあったとしても、類似したインシデントの情報を抽出することができる。

しかしながら、このシステムでは、新種のインシデントへの対応を考慮していない。発生したインシデントが、過去に蓄積したいずれのインシデントとも異なる新種のインシデントであった場合、このシステムでは、過去のインシデントのうち、最も類似したインシデントの情報が提示される。しかしながら、最も類似したインシデントの情報が、当該インシデントの対応に有用だとは限らず、オペレータが提示された情報に従って対処をすることにより、インシデントの初動対応がかえって遅れてしまうことさえありうる。

本稿では、この問題に対して、新種のインシデントに対しても、管理者に誤解させることなく、当該インシデント対応のためのヒントとなる情報を提示する方法について議論する。

2. インシデント情報蓄積・検索システムとインシデントに関する情報の提示方法

2.1. インシデント情報蓄積･検索システムの概要

本節では、本稿で検討するインシデント情報蓄積・検索システムについて述べる。本システムでは、機械学習によるエンジンとデータベースからなり、以下の処理を行う。

2.1.1. インシデント情報の蓄積

新たなインシデントが発生し、そのインシデントの特徴（異常検知されたフローの情報等）が得られた際には、その情報をシステムに投入する。システムに投入した際には､インシデントに対する ID を割り当て、ID と当該情報をデータベースに保存する。

2.1.2. インシデント対応情報の蓄積

インシデントの対応が完了した際には、オペレータは対応したインシデントの ID と合わせて、当該インシデントの種類名、インシデントの対応手順をシステムに投入する。システム側では、投入された情報をデータベースに保存する。

2.1.3. インシデント対応情報の提示

新たなインシデントが発生し、当該インシデントに対応するための情報が必要な場合、オペレータは当該インシデントに関する特徴量をシステムに投入する。システム側は、この特徴量をもとに、機械学習、データベースを組み合わせ、必要な情報をオペレータに提示する｡提示すべき情報については､次節で議論する。

2.1.4. 学習

定期的にデータベースに蓄積されたインシデント情報をもとに、システム内で利用しているモデルを学習する。学習が必要なモデルの構成や、そのモデルの使い方は、システムの構成によって異なる。本稿で用いた学習モデルについては、3 節で述べる。

2.2. インシデントに関する情報の提示方法

本節では、インシデント情報蓄積・検索システムにおいて、インシデントに関する情報をどのように提示するかを議論する。

既知のインシデントについては、我々が以前構築したように、当該インシデントの種別に関する情報や、その種別のインシデントに対して、過去に蓄積した対処方法の情報を提示することにより、オペレータの迅速なインシデント対応が可能となる[3]。しかしながら、新種のインシデントについては、対応するインシデント種別の情報も、対処方法の情報も蓄積がない。そのため、上述の情報の提示を行うことは不可能である。

また、本来は新種のインシデントであるにも関わらず、既知のインシデントであると誤った情報を提示してしまうことは避ける必要がある。誤った情報を提示し、オペレータがその情報を信用して、インシデントへの対応を行った場合、誤ったインシデント対応を行ってしまうことが考えられる。この場合、例えば、外部からの脆弱性診断が行われているのみであるという誤った情報が提示されたが、本来は、対処が必要なインシデントであった場合、インシデント対応が優先されず、インシデントへの対応がかえって時間を要することも考えられる。

そのため、新種のインシデントに対する情報提示で、重要なのは、既知のインシデントであるのか、新種のインシデントであるのかをオペレータが判断するのに必要な情報を提示できることである。本稿では、新種のインシデントであるかを判断するのに有用な情報として、既知のインシデントとの相違点を提示することを考える。これにより、オペレータは相違点を確認し、既知のインシデントと同種のインシデントであるのか、それとも異なるインシデントであるのかを判断することができる。

すなわち、本稿では、発生・検知されたインシデントについて下記の情報を提示するものとする。

- 類似する既知のインシデントのラベル名
- 類似する既知のインシデントの対処に関する情報
- 既知のインシデントとの相違点があれば相違点

3. インシデント情報を提示するシステム

我々は、2 節のインシデント情報蓄積・検索システムにおいて、2.2 節で議論した情報をオペレータに提示することができるシステムを構築した。本システムでは、インシデント情報を蓄積するデータベースの他に、機械学習により、インシデントの分類や、分類された既知のインシデント分類との相違点を出力する。その後、分類されたインシデント名で検索することにより、当該インシデントの対処方法の情報を得ることができる。

以降、本節では、インシデントの分類や、分類された既知のインシデント分類との相違点を出力するための機械学習モデルについて述べる。

3.1. 概要

図 1 にインシデント情報提示システムの概要を示す。本インシデント情報提示システムでは、まず、観測されたインシデントの特徴量から、ルールベースで特徴量を抽出する。抽出した特徴量をもとに、ニューラルネットワークベースの分類器で分類を行い、発生したインシデントが、既知のどの種類のインシデントに類似しているかを出力し、そのインシデント名を提示する。合わせて、分類されたインシデント名とルールベースで生成された特徴量をもとに、Conditional Variational Auto Encoder[4] を用いて、当該種類のインシデントであるとみなした上で、特徴量を生成する。そして、出力された特徴量と実際の特徴量を比較することにより、既知のインシデントとの相違点を出力する。

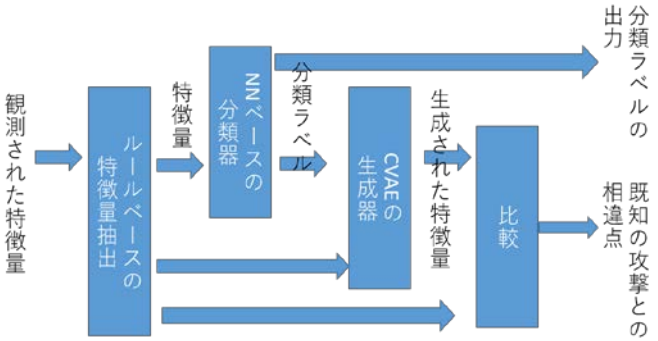

図 1　インデント情報提示システムの流れ

3.2. 特徴量抽出器

本稿では、インシデントに関する情報は、異常検知された通信フロー群について、情報を提示するものとし、各通信フロー群について、特徴量抽出器では、以下の特徴量を抽出するものとした。

- 送信元が既知か、否か
- 宛先が既知か、否か
- 上り方向の平均通信量（Byte）
- 下り方向の平均通信量（Byte）
- 上り方向の平均通信レート（Byte/sec）
- 下り方向の平均通信レート（Byte/sec）
- フロー継続時間
- 同一ポートに送られたフロー数
- 同一宛先に送られたフロー数
- 同一送信元から送られたフロー数

3.3. ニューラルネットワークによる分類器

本稿では、分類器は隠れ層の数が 2 の全結合型ニューラルネットワークとして構成した。入力は、前節で述べた特徴量抽出器の出力(10 次元)とし、出力は、各要素が該当する分類結果の分類確率を示すN次元のベクトル（Nは分類対象の要素数）とした。本ニューラルネットワークの学習の際には、誤差関数として Softmax Cross Entropy を用いた。

3.4. Conditional Variational AutoEncoder による生成器

本稿では、Conditional Variational AutoEncoder による生成器によって、元の特徴量を生成する。図 2 に本稿で用いた Conditional Variational AutoEncoder の概要を示す。

Conditional Variational AutoEncoder は、入力特徴量・分類ラベルから潜在変数Zを求める Encoder と、潜在変数Zと分類ラベルから特徴量を出力する Decoder からなる。なお、Encoder は、潜在変数Zを出力するのではなく、潜在変数Zは多変量正規分布Gaussian(μ, σ)に従うものとし、Encoder では、多変量正規分布のパラメータμ、σを出力するものとする。また、分類ラベルは、N 次元の one-hot ベクトルで表されるものとする。

Conditional Variational AutoEncoder の学習の際には、以下の誤差を最小化するように学習した。

$$(1-\mathrm{w})\mathrm{P}(\mathrm{x}^{\mathrm{in}},X^{out})+\mathrm{wR}(\mathrm{x}^{\mathrm{in}},X^{out})+\mathrm{CD_K}\big(\mathrm{N}(\mu,\sigma)\big|\mathrm{N}(0,\mathrm{I})\big)$$

ただし、本誤差は、入力特徴量x^{in}を与え、Encoder の出力であるμ、σを取得し、μ、σから潜在変数zを正規分布に従って複数生成し、zを入力とした Decoder の出力x^{out}を複数取得した際の誤差であり、X^{out}は出力x^{out}の集合である。$\mathrm{D_K}\big(\mathrm{N}(\mu,\sigma)\big|\mathrm{N}(0,\mathrm{I})\big)$は、正規分布$\mathrm{N}(\mu,\sigma)$と正規分布$\mathrm{N}(0,\mathrm{I})$の KL ダイバージェンスであり、学習における正則化項である。Cは再構成誤差への重みを定めるパラメータである。

また、$\mathrm{P}(\mathrm{x}^{\mathrm{in}},X^{out})$は、$\mathrm{x}^{\mathrm{in}}$が生成した$X^{out}$の範囲外となる場合のペナルティを示し、$\mathrm{R}(\mathrm{x}^{\mathrm{in}},X^{out})$は、生成した$X^{out}$と$\mathrm{x}^{\mathrm{in}}$の類似度を比べた再構成誤差である。次節で述べるように、本研究では、Conditional Variational AutoEncoder は、当該分類に該当したと考えた際の取りうる特徴量の範囲を求めるために用いる。そのため、x^{in}は生成された特徴量の集合X^{out}の範囲外にある場合は、大きなペナルティをかすのが適切と考えられ、本研究では、ペナルティ項$\mathrm{P}(\mathrm{x}^{\mathrm{in}},X^{out})$を導入している。

なお、本研究では、$\mathrm{P}(\mathrm{x}^{\mathrm{in}},X^{out})$は、以下のように定義した。

$$\mathrm{P}\big(\mathrm{x}^{\mathrm{in}},X^{out}\big)=\sum_{\mathrm{i}}\left(\left[\mathrm{x_i}^{\mathrm{in}}-\max_{\mathrm{n}}\mathrm{x}^{\mathrm{out}}_{n,i}\right]^{+}+\left[\min_{\mathrm{n}}\mathrm{x}^{\mathrm{out}}_{n,i}-\mathrm{x_i}^{\mathrm{in}}\right]^{+}\right)^{2}$$

ただし、$\mathrm{x_i}^{\mathrm{in}}$は、特徴量抽出器が出力したi番目の特徴量、$\mathrm{x}^{\mathrm{out}}_{n,i}$は$X^{out}$中の$n$回目の出力のi番目の特徴量であり、$[\mathrm{x}]^{+}$は$\mathrm{x}>0$の場合はx、それ以外は 0 とする。この定義より、x^{in}のすべての要素がX^{out}に含まれる特徴量の最大と最小の間にある場合は、$\mathrm{P}\big(\mathrm{x}^{\mathrm{in}},X^{out}\big)$=0 となり、$\mathrm{x}^{\mathrm{in}}$が$X^{out}$に含まれる特徴量の範囲から逸脱すればするほど大きな値となる。

また本研究では、$\mathrm{R}(\mathrm{x}^{\mathrm{in}},X^{out})$についても以下のように二乗誤差を用いて定義した。

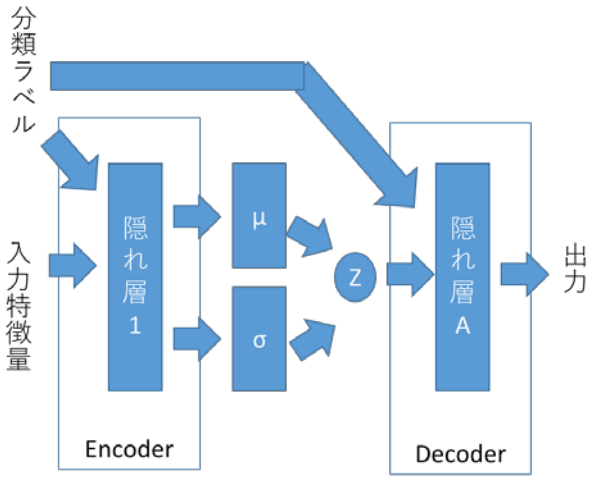

図 2 Conditional Variational AutoEncoder の概要

$$\mathrm{R}\big(\mathrm{x}^{\mathrm{in}},\mathrm{X}^{\mathrm{out}}\big)=\frac{1}{|\mathrm{X}^{\mathrm{out}}|}\sum_{i,n}\left(\frac{\mathrm{x_i}^{\mathrm{in}}-\mathrm{x}^{\mathrm{out}}_{n,i}}{\mathrm{x_i}^{\max}-\mathrm{x_i}^{\min}+\epsilon}\right)^{2}$$

ここで、ϵは十分に小さな定数であり、$\mathrm{x_i}^{\max}$は学習に用いた全サンプルにおけるi番目特徴量の最大値、$\mathrm{x_i}^{\min}$は学習に用いた全サンプルにおけるi番目特徴量の最小値である。$\mathrm{x_i}^{\max}-\mathrm{x_i}^{\min}+\epsilon$で$\mathrm{x_i}^{\mathrm{in}}-\mathrm{x}^{\mathrm{out}}_{n,i}$を割った誤差について考えることにより、本研究では、各特徴量の取りうる範囲を考慮して、各特徴量の誤差に重みを付けている。本誤差は、出力として得られるX^{out}が入力x^{in}から離れるほど大きくなる。そのため、本誤差関数の値を小さくするような学習を行うことにより、各分類ラベルに合わせて、入力特徴量に近い特徴量を出力することができる。

3.5. 比較器

本稿では、説明を出力する際には、Conditional Variational AutoEncoder により、M個の特徴量を出力し、その出力された特徴量と、特徴量抽出器により抽出された特徴量を比較する。そして、以下のルールにより、既知のイベントと異なる特徴について出力する。

- $\mathrm{x_i}^{\mathrm{in}}>\max_{\mathrm{n}}\mathrm{x}^{\mathrm{out}}_{n,i}$の場合
  i番目の特徴量は、既知のものより大きい
- $\mathrm{x_i}^{\mathrm{in}}<\min_{\mathrm{n}}\mathrm{x}^{\mathrm{out}}_{n,i}$の場合
  i番目の特徴量は、既知のものより小さい
- 上記以外
  i番目の特徴量については、相違点を出力しない

上記のルールにより、既知のインシデントと比べ、明らかに異なる特徴量について、その特徴量が既知のものと比べ、大きい、あるいは、小さいかを出力

表 1 ：提示された情報

正解となる分類名	判別結果
SYN flood	SYN flood　(相違点なし)
Port scan	Port Scan（相違点なし）
Brute force attack	Port Scan （ただし、 上り方向の通信量(Byte)、上り方向のパケット数、 上り方向の通信レート(Byte/sec)、 上り方向の通信レート(Packets/sec) 、 下り方向の通信量(Byte)、下り方向のパケット数、下り方向の通信レート(Byte/sec)、 下り方向の通信レート(Packets/sec)、フローの継続時間は、port scan よりも大きい）

することができ、オペレータに対して、既知のインシデント分類にマッチしているかを判断することができる情報を提示することができる。

4. 動作確認

我々は、3 節で述べたシステムを、Chainer を用いて実装し、動作確認を行った。本節では、動作確認内容、動作確認結果について述べる。

4.1. 動作確認に用いるデータ

本稿では、同一サーバ上に、ペネトレーション用 Linux ディストリビューションである KaliLinux[5]をインストールした複数の仮想マシンを動作させた。その上で、以下の攻撃を生成し、仮想マシン間に流れるパケットをキャプチャしたものを用いる。

SYN flood: 送信元を偽装した SYN パケットを同一宛先に大量に送信することにより、システムを過負荷に陥らせる攻撃。本稿では、HTTP サーバに対して、攻撃を加えた。

Port Scan: 宛先のシステムで動作しているポート番号を確認すること。本稿では、宛先にサーバに対して、SYN パケットを送信し、その応答を確認することにより、開きポートを確認する SYN スキャンを行った。

Brute force attack: パスワード認証がかかっているシステムに対して、当該システムのパスワードを窃取することを目的とし、ユーザ名、パスワードを総当たりで試す攻撃。本稿では、HTTP Digest 認証を行っているシステムに対して、パスワード総当たり攻撃を行った。

4.2. 動作確認のシナリオ

本稿では、特に、未知の攻撃が発生した際に、その攻撃に対する情報を正しく提示できるかということに焦点をおいている。そのため、動作確認の際にも、上述の 3 種類の異常のうち、一つは、未知の異常であるという状況下での評価を行う必要がある。本稿では、SYN flood、Port Scan は既知で、Brute force attack は未知という状況において、各異常が与えられた際の出力について確認した。

4.3. 動作確認結果

表 1 に提示された情報を示す。表より、すでに同種のインシデントが学習されている SYN flood、Port scan については、それぞれ、SYN flood、Port Scan と正しく判別されており、また、相違点もなしと出力されている。そのため、この出力を得たオペレータは、発生したインシデントが SYN flood、Port Scan であると認識することができる。

それに対して、未知の異常である Brute force attack に対しては、既知の異常の中では、Port scan に似ていると出力されている。ただし、Port scan との相違点として、通信量、パケット数、通信レート、フローの継続時間が出力されている。Port scan と比べ、Brute force attack では、実際にサーバに接続し、パスワードを送信し、その応答を確認しているため、通信量、パケット数、通信レート、フローの継続時間は大きくなる。つまり、本システムは、正しく port scan との相違点を出力することができている。また、オペレータはこの出力を見ることにより、既知の異常とは違う新種の異常であると判断することができ、異常に関する手がかりとなる情報も得ることができる。

5. まとめ

インシデントが発生した際には、管理者はそのインシデントに対して、迅速に対応することが求められる。攻撃に対して迅速な対応を行うためには、異常検知技術を用い、インシデントを迅速に検出するだけでなく、そのインシデントについて分析することが必要となる。

我々は、これまでに、過去に発生したインシデント情報を蓄積し、新たにインシデントが発生した際に、過去の類似インシデントを検索し、その情報を提示するシステムを構築してきた。しかしながら、このシス

テムでは、新種のインシデントへの対応を考慮しておらず、新たに発生したインシデントが、過去に蓄積したいずれのインシデントとも異なる新種のインシデントであった場合、このシステムでは、過去のインシデントのうち、最も類似したインシデントの情報が提示するのみで、その情報が当該インシデントの対応に有用だとは限らず、オペレータが提示された情報に従って対処をすることにより、インシデントの初動対応がかえって遅れてしまうことさえありえた。

本稿では、この問題に対して、新種のインシデントに対しても、管理者に誤解させることなく、当該インシデント対応のためのヒントとなる情報を提示する方法について議論した。本稿で議論した手法では、新たなインシデントが発生した際には、過去の類似した事象の名前を提示するとともに、類似事象との相違点がある場合には、相違点についても提示する。本稿では、本システムを実装し、正しく情報提示できることを示した。

謝辞

本研究の一部は，内閣府が進める戦略的イノベーション創造プログラム(SIP)「重要インフラ等におけるサイバーセキュリティの確保」(管理法人：NEDO)によって実施されました。

文　献

[1] H. Sinanovic and S. Mrdovic, “Analysis of mirai malicious software,” in Proceedings of SoftCOM, Sept. 2017

[2] 中津留毅, 五十嵐弓将, 南拓也, “IoT 機器の健全性劣化を検知する技術,” 電子情報通信学会総合大会, Mar. 2017.

[3] 大下裕一, 村田正幸, "特徴量写像関数の学習による類似インシデント検索," 電子情報通信学会技術研究報告(IN2017-57), pp. 67-72, Dec. 2017.

[4] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in Proceedings of Advances in neural information processing systems, pp. 3581-3589, 2014.

[5] https://www.kali.org/