Vol.2017-DPS-170 No.21
Vol.2017-CSEC-76 No.21
2017/3/3

プロトコルフォーマット推定処理の高速化手法の提案

木藤 圭亮† 山本 匠† 西川 弘毅† 河内 清人†

概要：プロトコルフォーマット推定技術は、未知のパケットデータセットを入力とし、頻度分析などの統計処理を主として、未知のプロトコルフォーマットの推定を行う。統計分析の際に有意な差が出にくいプロトコルの場合は、推定に多くの時間を要していた。一方で制御ネットワークは一定周期時間ごとに通信を行う、周期通信が特徴的であり、通信周期は制御対象によって固有である。本発表では、事前にパケットデータセットを周波数解析することで、パケットを通信周期ごとに分類し、パケットフォーマット推定処理を高速化する手法を提案する。

キーワード：パケットフォーマット推定、パケット分類、制御システムネットワーク、C&C 通信、高速化

Proposal on Speeding-up Automated Protocol Format Inference

Keisuke Kito† Takumi Yamamoto† Hiroki Nishikawa† Kiyoto Kawauchi†

Abstract: Protocol Format Inferring requires input as unknown packet dataset and output inferred protocol formats. Statistical processing such as frequency analysis is dominant work in Protocol Format Inferring. In case of complicated protocols such as transferred in binary formats, it spend more amount of time in statistical works than simple protocols. In this paper, we propose speed-up method for Protocol Format Inferring that classifying packet cyclic time using frequency analysis technique before format inferring process.

Keywords: Protocol Format Inferring, Packet classifying, Control System Networks, C&C Communication, Speeding-up

1. はじめに

工場や発電所で用いられる、産業用制御システムに対するサイバー攻撃事例が近年多発している。産業用制御システムにおける通信プロトコルは、近年標準化が進んではいるものの、未だ独自プロトコルが用いられている場合が多い。またマルウェアの C&C サーバとの通信でも、攻撃者は独自のプロトコルを設計して使用する場合が多い。独自プロトコルの多くは仕様が公開されておらず、問題に行き当たった場合は通信内容を解析する必要がある。このような未知のプロトコルを解析する必要に迫られた場合、その作業は人間が多くの作業量と時間をかけて実施しているのが現状である。

自動プロトコルリバースエンジニアリングは、未知のプロトコルの通信データや通信プログラムを入力として、通信プロトコルの仕様を推定する技術である。推定したプロトコル仕様は、IDS のルール記述や C&C 通信の解析、さらにはファジングのルール生成などへの応用が期待できる。しかし、仕様推定は頻度分析などの統計処理が主であり、未知の大量のパケットデータから推定するため、処理に多くの時間を要するのが問題であった。

一方で産業用制御システムネットワークでは、対象システムを制御するために、センサー等で計測した物理値をフィードバックし、その物理値に応じたアクチュエータ操作を一定時間ごとにネットワークを介して実施する。このときの通信周期は対象システムの特性に応じて、適切な値が設定される。

本稿では、産業用制御システムネットワークなどにおける、一定時間周期ごとに行う通信、いわゆる周期通信に着目し、周波数解析のテクニックを応用して、パケットを通信周期ごとに予め分類を行うことで、パケットフォーマット推定処理の高速化を行う手法を提案する。

2. 自動プロトコルリバースエンジニアリング

自動プロトコルリバースエンジニアリング(Automatic Protocol Reverse Engineering、以下 APRE)は、仕様が不明なプロトコルの通信データや通信プログラムなどを入力とし、プロトコル仕様に関する情報を推定する技術である。具体的には、キャプチャした通信パケットデータや、通信を行うプログラムバイナリなどを入力とする。APRE に関する Narayan ら調査論文[1]では、推定するデータで大きく 2 つに分類している。1 つは通信パケットデータのフォーマットを推定するものであり、もう 1 つは通信プロトコルの状態遷移を推定するものである。

APRE は 2004 年に Beddoe らの PIP(Protocol Informatics Project)[2]によって研究が加速し、現在では様々なパケット

† 三菱電機株式会社 情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

Vol.2017-DPS-170 No.21
Vol.2017-CSEC-76 No.21
2017/3/3

フォーマット推定手法が提案されている。当初は生命情報学(Bioinformatics)における、遺伝子の塩基配列分析アルゴリズムの一つである Needleman-Wunsch Algorithm[6]を、ネットワークパケットのフォーマット推定に応用したものであった。Beddoe らの研究以降、APRE の研究は盛んになり、自然言語処理を応用し精度向上を行う方式[3]、通信プログラムバイナリをテイント解析し推定する方式[5]、頻度分析などの統計処理を繰り返し実施するもの[4]などが先行研究として挙げられる。

推定したプロトコルフォーマットは、仕様が未知なプロトコルを検知するための IDS/IPS ルールの作成や、Fuzzing を効率的に行うためのファジングルールの生成、さらにはマルウェアの C&C サーバ通信に独自プロトコルが用いられて場合は、その通信プロトコルの解析に応用することができる。

次に推定するデータによる分類について、詳しく述べる。

2.1 状態遷移推定型

状態遷移推定型では、あるプロトコルを用いて通信を行う際に通信ホストそれぞれが持つ、通信状態の状態遷移パターンを推定する。TCP 通信の LISTEN や ESTABLISHED などが具体例であり、このような通信プロトコルの状態遷移を通信パケットデータセット、通信プログラムのバイナリ等を用いて推定する。本稿では状態遷移推定型は対象としない。

2.2 パケットフォーマット推定型

パケットフォーマット推定型では、パケットデータ中に含まれる各フィールドのデータ長、境界、データ形式、エンディアンなどについて推定を行う。例えば HTTP では、データのヘッダ部とボディ部の間には空行が挿入されており、これをデータの境界とみなすことができる。またヘッダ部にはリクエストが書かれており、例えば「`GET / HTTP/1.1`」という HTTP リクエストについて、複数のパケットデータを解析すると、データ形式は文字データでやり取りが行われており、「`GET`」の部分は他に POST など数個のバリエーションがあること、「`HTTP/`」は固定文字列であることなどの、データ形式やデータ長に関する情報が推定できる。上記の事例を図 1 に示す。

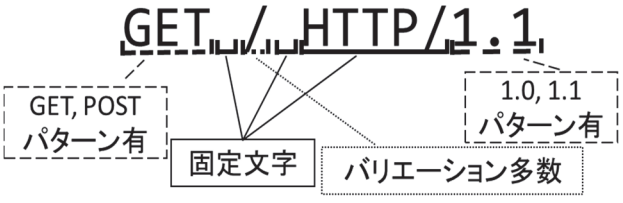

図 1　HTTP リクエストの解析例

図 1 のように推定したパケットフォーマットは図 2 のような有効グラフとして書き下すことができる。ここで*はバリエーションが多いため、ワイルドカードであることを示している。

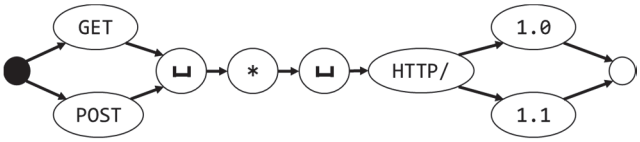

図 2　推定したフォーマットの表現例

また解析対象とするプロトコルの形態で解析の困難性が変わる。今回は通信内容が文字データとしてやり取りが行われるテキストベース通信と、通信内容をバイナリでやり取りを行うバイナリベース通信で比較する。テキストベース通信では文字列のみを扱っているため、出現頻度で有意な差が出やすいかつ、自然言語処理の手法が応用できる。また各データフィールドを区切るために、空白文字や改行コードが挿入されるため、一般的に解析が容易である。一方でバイナリベース通信では、テキストベース通信に比べて、バイナリ値の全てが値域であるため、頻度分析で有意な差が出にくい。また、予めプロトコルとして区切りのバイト位置を決めてある場合が多く、フィールドの区切り位置を見つけることが困難である。そのためバイナリベース通信では、統計解析に多くの時間を費やしてプロトコルフォーマット推定を行う。表 1 にテキストベース通信とバイナリベース通信の特徴についてまとめた。

表 1　各データ表現形式の特徴

テキストベース(Ex. HTTP)	バイナリベース(Ex. SMB)
○頻度の偏りが出やすい ○自然言語処理が使用可 ○区切り文字がある	○短いデータ長で伝送可 ○リアルタイム性
×データ長が長くなる ×文字コード整合の問題	×頻度に偏りが出にくい ×区切り文字がない

パケットフォーマット推定は入力データによって 2 つに分類することができる。1 つは、通信パケットデータを入力とするネットワークトレース型、もう 1 つが通信プログラムバイナリを入力とする、バイナリトレース型である。

ネットワークトレース型は複数のパケットデータで構成されるパケットデータセットを入力とする。pcap 形式のデータが具体例にあたる。ネットワークトレース型は統計解析や自然言語処理などのデータ解析技術を核としてパケットデータセットのデータを解析するものである。しかしパケットデータセットに、推定したいプロトコルの全てのバリエーションが含まれているとは限らないので、推定に漏れがある可能性がある。

バイナリトレース型は推定したいプロトコルで通信を

Vol.2017-DPS-170 No.21
Vol.2017-CSEC-76 No.21
2017/3/3

行うプログラムのバイナリを入力とする。ソフトウェア内部に通信の仕様を落とし込んで実装されており、内部状態の分岐条件などがプログラムバイナリに含まれるため、一般的にネットワークトレース型に比べて高い精度で推定が可能である。ソフトウェアバイナリを入力とするので、ソフトウェアリバースエンジニアリング技術を核とするものである。

本研究ではネットワークトレース型のパケットフォーマット推定を対象とする。

3. 制御ネットワークとフォーマット推定

3.1 産業用制御ネットワーク

産業用制御ネットワーク（以下、制御ネットワーク）は、システム制御のために設計されたネットワークであり、主に工場の自動制御装置(FA 機器)や、発電プラントなどに用いられている。一般の情報ネットワークとの違いは、主にリアルタイム性があること、耐故障性を持つことである。リアルタイム性については、システム制御において必須である周期通信の機能や、データ送信要求から一定時間以内にデータの到着が保障する機能を有する。耐故障性では、通信部分を 2 重系にし、故障してもリアルタイムに切り替える機能を持っている。このように制御ネットワークではリアルタイム通信を必須とするので、その多くはバイナリベース通信でやりとりが行われる。

さて一般的なシステム制御は、センサーで対象システムの物理値を計測し、物理値を目標値に合わせるようにアクチュエータを操作する、いわゆるフィードバック制御が行われている。フィードバック制御では、計測と操作を一定周期ごとに行う必要があり、制御システムネットワークでは、パケットの伝送をリアルタイムに行うことで、それを実現している。データの送受信を行う周期は、制御対象となるシステムによって固有である。これは制御対象ごとに物理値として反映される速度や、アクチュエータの操作に対する応答速度が異なるためである。逆に言えば、制御対象システムごとに通信周期が異なるので、通信周期が異なれば通信の目的・内容が異なると考えることができる。つまり、通信周期でパケットを分類することは、通信目的でパケットを分けることに対応付けられることを意味する。

3.2 パケットフォーマット推定の課題

制御ネットワークの多くはバイナリベースで行われるが、バイナリベース通信のプロトコルフォーマット推定では、精度を向上させるために、統計処理を繰り返し実施する。例えば Wang らの Biprominer[4]は、バイナリベース通信をネットワークトレース方でプロトコルフォーマットを推定するものである。Biprominer では、パケットデータをバイト単位で頻度分析を行い、頻度が高かったバイトを含む 2 バイトのバイト列で更に頻度分析を行う、これを数回繰り返し、出現頻度の高い n バイト長のバイト列を得る。これをラベル付けし、ラベル付けされていないバイト列のみで、同様の処理を繰り返す。これを全てのバイト列がラベル付けされるまで繰り返す。膨大なパケットデータセットをバイト単位で処理を行うため、推定処理には膨大な時間を要する。Biprominer の処理の流れを図 3 に示す。

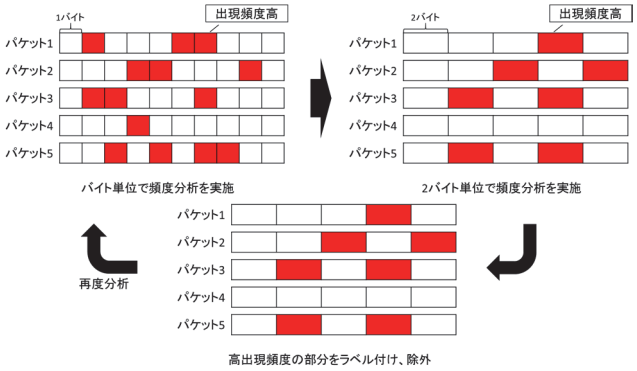

図 3 Biprominer[4]の推定処理の概要図

4. 提案手法

提案手法のアイディアは、ある周期で通信を行うパケットが同じ目的を持った通信であるという前提のもとで、通信周期でパケットを分類することにより、統計分析のときに有意な差が生まれやすくすることで、パケットフォーマット推定処理を高速化するものである。通信周期で分類する手法として、パケットデータセット内のパケットデータとパケット到着時刻を利用して時系列データを構成し、周波数解析を行うことで、パケットを周期時間ごとに分類を行う。以後、提案手法を 3 つのフェーズに分けて説明する。

4.1 時系列構成フェーズ

はじめにパケットデータセットから時系列データを構成するフェーズ、時系列構成フェーズについて説明する。

パケットデータセット、例えば pcap 形式のパケットキャプチャファイルには、パケットデータのほかにパケット到着時刻の情報が含まれている。時系列構成フェーズでは、パケットデータと到着時刻を対応付けることで時系列データを構成する。しかし各パケットのデータ長はパケットごとに異なるため、時系列データを構成するためにはパケットデータを加工する必要がある。今回はパケットデータの先頭から、ユーザが指定したバイト長だけを取り出し、そのバイナリ値そのものを時系列データの振幅とする手法を提案する。一般的にデータ列は先頭部分のデータ、いわゆるヘッダ情報に相当するデータが、後方部分データ、つまりデータ本体部分の属性や情報を表す。また同じ周期通信のパケットは同じ目的の通信であるという前提なので、同じ通信周期のヘッダ値は類似したバイナリ値になる可能性

Vol.2017-DPS-170 No.21
Vol.2017-CSEC-76 No.21
2017/3/3

が高い。そのため、パケットの先頭部分の一部を抽出することが望ましい。時系列構成フェーズの概要を図 4 に示す。

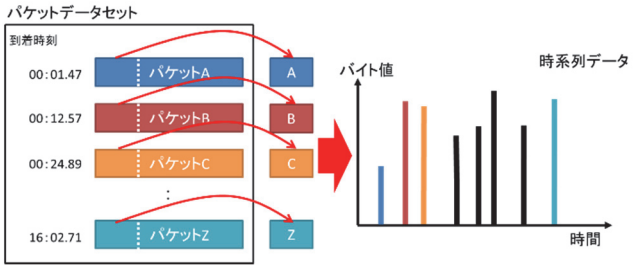

図 4　時系列構成フェーズの概要

4.2 周波数解析フェーズ

次に周波数解析フェーズを説明する。周波数解析フェーズでは 3 つの処理に分けられる。はじめに、時系列構成フェーズで構成した時系列データをフーリエ変換して、周波数データを生成する周波数変換処理。次に、周波数データから、抽出したい通信周期に相当するスペクトルのみを取り出し、それ以外をゼロにセットする周期スペクトル抽出処理。最後に、抽出後の周波数スペクトルを逆変換して、時系列データに戻す周波数逆変換処理である。

周波数変換処理では、時系列構成フェーズで構成した時系列データを入力として離散フーリエ変換を行う。このとき高速フーリエ変換を用いることで高速処理が可能である。

次の周波数抽出処理でははじめに、変換して得られた周波数スペクトルをパワースペクトルとして表現する。次にユーザが予め設定した、抽出したい通信周期に相当するスペクトルが存在するかどうかを確認する。これは例えばスペクトルの平均値に比べて、指定した部分のスペクトルが大きい場合に存在すると判断する形で実現できる。この処理を抽出したい周期ごとに行い、周期時間ごとの抽出後周波数スペクトルが生成される。

最後の周波数逆変換処理では、抽出後周波数データをフーリエ逆変換することで、抽出したい周期通信パケットのみが残存する、抽出後時系列データを得ることができる。これを抽出した周波数スペクトルの数だけ逆変換を行い、各々の通信周期に対応する抽出後時系列データが得られる。

ここまでが周波数解析フェーズの一連の処理の流れであるが、これが行える原理として、振幅A、周期Tの周期的デルタ関数(コム関数とも呼ばれる)を理想的な周期通信のみの時系列データとして考える。これをフーリエ変換すると、以下のような式が得られる。

$$\mathcal{F}(\mathrm{A}\delta(t-n\mathrm{T})) = \mathrm{A}\frac{\sqrt{2\pi}}{\mathrm{T}}\delta\left(\omega - n\frac{2\pi}{\mathrm{T}}\right)$$

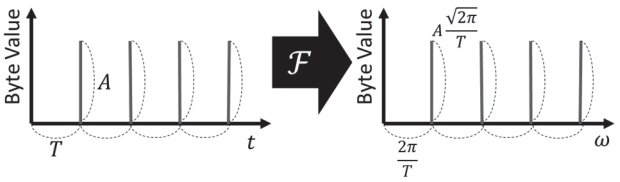

図 5　周期的デルタ関数のフーリエ変換

ただしnは整数である。図式化すると、図 5 のようになる。図 5 を見ると、周期的デルタ関数を周波数変換すると、周期が$\frac{2\pi}{T}$倍、振幅$\frac{\sqrt{2\pi}}{T}$倍でスケールしたうえで、周波数領域においても周期的デルタ関数の形を保っていることが分かる。そのため予め周期Tで抽出すると分かっている場合、周波数領域の特定部分にスペクトルが出ているかどうかを見ることで、周期Tで周期通信が行われているかどうかを確認することができる。周期Tの周期的デルタ関数に相当する周波数スペクトルのみを残した、抽出後周波数データを逆変換することで、所望の周期Tのパケットに相当するデータのみが残る抽出後時系列データを得る。

周波数抽出フェーズにおいて、指定した周期に相当するスペクトルが平均値よりも小さい場合は、抽出に失敗したか、設定した周期時間に関する周期通信が行われていないことと判断する。

4.3 パケット関連付けフェーズ

パケット関連付けフェーズでは、得られた抽出後時系列データの振幅・到着時刻の情報をもとに、入力したパケットデータセットを対応付けることで、周期通信を行うパケットを取り出すことができる。これらの処理を周波数解析フェーズで得られた抽出後時系列データの数だけ実施する。結果として、分類した通信周期の数だけの分類後パケットデータセットを得る。パケット関連付けフェーズの概念図を図 6 に示す。

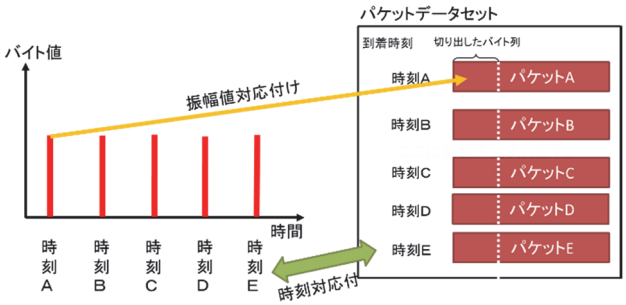

図 6　パケット関連付けの概要

ここで抽出後時系列データの振幅と時刻に着目して、もとのパケットデータセットと対応付けを行うと説明したが、到着時刻と振幅は必ずしも一致するとは限らない。なぜな

らば、周波数抽出フェーズにおいて、通信周期に関連するスペクトル以外の成分を削除しているため、逆変換した際にもとの時系列データの情報を完全に復元できないためである。そのため、時刻、振幅で対応付ける際には、予めユーザが指定した誤差範囲に収まるものを、対応するパケットとして関連付けを行う。

最終的に行いたいのはパケットフォーマット推定処理であるので、通信周期時間ごとに分類した後、パケットデータセットごとにフォーマット推定処理を実施し、推定パケットフォーマットを得る。

5. 考察

5.1 通信周期時間ごとに分類することの妥当性

先述したとおり、制御システムネットワークにおいては、一定周期で通信を行い、センサーで計測した物理値のフィードバックと、物理値に応じたアクチュエータ操作を制御ネットワーク経由で行う。通信周期は制御対象となるシステムごとに固有の周期を持つので、ある周期の通信は、それに対応したシステムを制御するための通信である可能性が非常に高い。そのため、通信周期ごとにパケットが分類できたとすると、通信周期ごとに分類されたパケットにおいて頻度分析を行った場合、有意な差がでやすいと考えられる。

また非周期通信を考慮していないが、非周期通信は通信周期で分類したときにどこの周期にも属さない可能性が高い。万が一、ある周期通信に分類されたとしても、低い出現頻度として有意な差が生まれると推測される。よって、周期通信を特徴とするプロトコルにおいては、通信周期時間ごとにパケットを分類することで、統計分析が行いやすくなると考えられる。

5.2 時系列データで周期通信の振幅を一定にする方法

周期通信のパケットに相当する情報を、周期的デルタ関数として時系列データを構成できることを前提としているため、時系列データを構成したときに、周期通信の振幅が一定であることも、本手法が機能する前提となっている。単純にバイト列を取り出してバイト値とする手法では、MSB に対して敏感になってしまう。つまり似たバイト列であっても MSB が異なるだけで、取りうる最大値の 2 分の 1 が、バイト値の差として生じる。これを回避する手法としていくつか提案する。

① 各パケットから取り出す箇所を適応的に変更する

周波数解析を行うときに、周期通信に相当するスペクトル成分が抽出できなかった場合に、バイト値を切り出す位置を変更して、時系列データ構成フェーズから再度実施する方法である。パケットデータセットから取り出す部分を変更する。このとき、各パケットで取り出す箇所は同一位置とするのが望ましいと考えられる。パケットの目的種別にあたるフィールド、例えば TCP 通信におけるポート番号に相当する部分が抽出できた場合、一定周期かつ一定振幅でバイナリ値が取り出すことができ、周波数解析処理が機能すると考えられる。周波数解析処理で抽出ができるまで、パケットの取り出す位置を適応的に変更することで、周波数解析が行えると考えられる。

② ハミング距離の値を振幅として用いる手法

各パケットから取り出したバイト列に対して、ある基準となるバイト列とのハミング距離を計算する。似たバイト列であれば、ハミング距離も似た値になる。だが抽出したい周期通信のバイト列を基準バイト列としたときは、取り出したい周期通信のバイト列とのハミング距離はゼロに近くなる。この場合は、バイト値がとりうる最大値から計算したハミング距離を新たなバイト値として定義することで、似たバイト列のみが取り出しやすくなる。そのため、ハミング距離を用いる方法では、ハミング距離そのものを振幅として用いる場合と、最大値からハミング距離の値を差し引いたものを振幅として用いる場合の 2 パターンを考える必要がある。上記 2 パターンについて試行し、どちらも周波数抽出フェーズで抽出が実施できなければ、基準バイト列を適応的に変更することで、周波数解析が実施できると考えられる。

5.3 制御ネットワーク以外のプロトコルへの適用可能性

今回は制御ネットワークのように一定時間ごとに通信を行うプロトコルに適した高速化手法を提案したが、一般の情報ネットワークで用いられるプロトコルで、提案手法はどのように機能するだろうか。例えば IP 電話のようなリアルタイムな通信を要求するプロトコルでは、時間周期性が出る可能性が高いため、提案手法が有効であると考えられる。一方でユーザの要求によって通信が行われるプロトコルでは、通信周期性が出にくいと考えられるため、提案手法でパケットを分類できないかつ、分類できたとしても、推定時の統計解析で有意な差がでないと考えられる。

6. まとめ

本稿では制御システムネットワークで用いられるような、一定時間ごとに通信を行うことを特徴とするプロトコルを対象として、周期時間ごとにパケットを分類することで、統計分析のフェーズで有意な差が出やすくなることを利用し、プロトコルフォーマット推定処理を高速化する手法を提案した。今後の課題として、周期時間で分類する手法の確認と、実際の制御ネットワークなどで用いられる周期通信に特徴のあるプロトコルを対象に、提案手法が効果的で

あるかを確認し、プロトコルフォーマット処理がどの程度高速化できるかを確認する。

参考文献

[1] John Narayan et al., A Survey of Automatic Protocol Reverse Engineering Tools, ACM Computing Surveys, Vol.48, No.3, Dec. 2015
[2] Marshall A. Beddoe, Network Protocol Analysis using Bioinformatics Algorithms, 2004, http://www.4tphi.net/~awalters/PI/pi.pdf (参照 2017/02/02)
[3] Joao Antunes et al., Reverse engineering of protocols from network traces, In 18th Working Conference on Reverse Engineering (WCRE), 2011
[4] Yipeng Wang et al., Biprominer: Automatic mining of binary protocol features, In 12th Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2011
[5] Juan Caballero et al., Polyglot: Automatic extraction of protocol message format using dynamic binary analysis. In 14th ACM CCS'07, 2007
[6] Saul B. Needleman et al., A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology. 48 (3): 443–53, 1970
[7] Netzob: Reverse Engineering Communication Protocols, https://www.netzob.org/ (参照 2017/02/02)