メタデータ作成によるマルウェア通信解析手法の提案

榊　辰哉†　水谷　正慶‡　武田圭史‡　　村井純†

†慶応義塾大学 環境情報学部

‡慶応義塾大学大学院　政策・メディア研究科

252-8520 神奈川県藤沢市遠藤 5322

{rekcah, mizutani, keiji, jun}@sfc.wide.ad.jp

あらまし　マルウェアの動的解析において，マルウェア通信を監視収集した通信データ群から，特定の特徴を持った通信データを効率的に抽出するためには，解析に必要となる情報を事前に処理する方法が有効である．本稿では，通信データから抽出したメタデータを格納するデータベースの構築と解析手法について述べる．このデータベースの利用により，収集した通信データ群からセッション毎の通信量や使用通信ポート等の特徴を持つ通信データを効率的に選び出すことができ，マルウェア動的解析が促進されると考える．また，CCCDATAset 2009を用いて，本手法の有効性を検証した．

A Proposal of an Analysis Method of malware communication by generating meta data

Tatsuya Sakaki†　Masayosi Mizutani‡ Keiji Takeda‡　Jun Murai†

†Keio University – Faculty of Environment and Information Studies

‡Keio University – Graduate School of Media and Governance

5322，　Endo，　Fujisawa-shi，　Kanagawa 252-8520，　Japan

Abstract In dynamic analysis of malware，preprocessing of communication data is effective to extract communication data with specific feature．In this paper，we describe database for storing meta data extracted from communication data for malware analysis．We can select communication data with specific feature by using this database，and it is expected to enhance productivity of researches on dynamic analysis．We tested the effectiveness of this method by applying it to CCC DATAset 2009.

1　はじめに

マルウェアの解析で多くの情報を得るためには検体の情報を直接解析する静的解析が望ましく，そのための研究[1]も進んでいるが，数多くのマルウェアを効率的に解析するにはマルウェアの動的解析も必要となる．動的解析において収集されたマルウェアの通信データは，事後的に一括して解析する場合が多い．この場合，収集した通信データ群は保存され，必要に応じて特定の通信データを取り出し解析する．しかし，収集した通信データが莫大な量になると，特定の通信情報を持つ通信データを探すために全ての通信データを解析し，検索しなくてはならないため効率が悪く，解析作業全体にかかる時間が増えてしまう．効

率化を図るためには，　解析の手がかりとなる情報を事前に処理し蓄積しておく方法が有効である．本稿では，通信データから解析の手がかりとなる情報をメタデータとして抽出し，格納するデータベースの構築と解析手法について述べる．通信データを選択する際に，使用しているTCP/UDPのポート番号やセッション毎の通信量などの特徴が事前に得られれば，解析者が必要としている通信データの発見が容易になり，効率的に解析を行えるようになる．このような仕組みを実現するため，通信データからメタデータを抽出するプログラムとメタデータを格納するデータベースを実装した．メタデータの特徴からデータベースを検索する事で，マルウェアの解析，研究の促進が期待される．

2　問題点

本稿でとりあげる問題は，マルウェア通信を監視して収集した通信データ群を解析する際に，未整備の膨大な通信データ群から解析すべき通信データを特定する作業の効率化である．筆者らは2008年7月から2009年5月の間マルウェアの動的解析を実施し，それぞれ20分間マルウェアを動作させ通信データを保存した．これによって，41,812個の通信データが生成され，データサイズは1.195TBであった．これらの通信データ群から解析すべき通信データを特定することは困難である．

これは，収集した通信データを保存したファイル（以下，通信データファイル）に何らかの解析をかけなければ通信データファイル内の情報が得られないことに起因する．従ってこの問題は，通信データに関する情報を抽出する前処理を行うことで解決される．

3　解決方法

3.1　要件定義

この問題の解決は通信データファイルに前処理を行うことであるが，具体的な解決方法は以下の4点の条件を満たしていなければならない．

(1)　処理の自動化

処理は解析の前処理であり，効率化を図るためには手動による処理は避けたい．よって，自動的に実行できる処理が望ましい．

(2)　処理結果の保存

処理の結果得られる情報は処理を行った直後にのみ必要となる訳ではないため，得られた情報を保存することで後から異なる条件で通信データを特定できる必要がある．その為にはある程度の保存領域が必要である．

(3)　通信データの特徴を捉えた結果

処理前には得られなかった通信データの特徴が，処理の結果得られることが望ましい．

(4)　検索可能な結果

処理の結果得られた通信データの情報群は研究者が一度に把握でき，任意の特徴を持つ通信データとそうでない通信データを選別できることが望ましい．

3.2　メタデータを格納するデータベースの提案

以上の4つの条件を満たす解決方法として，通信データから抽出したメタデータをデータベースに格納するという方法を提案する．

データベースは情報を保存し，それらの情報群から任意の条件を満たす情報を検索できる．よって，条件(2)，(4)を満たす．データベースを使用する場合，通信データのパケット毎に詳細な情報を保存すると情報が煩雑になり，処理前の通信データを解析するのと同様

の作業が必要となってしまう．従って，条件(3)を満たすために，データベースに保存する情報は全体のパケット数やセッション毎のデータなど，個々のパケット情報を統合して作成された通信データの特徴であることが望ましい．本稿ではこのような通信データの特徴をメタデータと定義する．メタデータは個々のパケット情報の統合であるため，単純なアルゴリズムによって通信データから抽出できる．よって， 条件(1)を満たす．よって，本稿では通信データから抽出したメタデータを格納するデータベースの構築を提案する．

4　データベースの構造

4.1　データベース概要

本稿ではマルウェアの通信データから抽出したメタデータを格納するデータベースを示す．本データベースにはマルウェア通信データの特徴となるメタデータを格納する．解析支援のために，メタデータには通信データ全体のパケット数などの特徴だけでなく，同じIPアドレスの組み合わせによる通信データ（フロー）毎の特徴，また同じフロー内で発生した同じTCP/UDP通信ポート番号の組み合わせによる通信データ（セッション）毎の特徴が含まれていることが望ましい．これらのメタデータを同じデータベース内に保存する場合，データベース内の別のテーブルに格納する必要がある．

4.2　テーブル構成

上記の条件に基づき作成したデータベースの構成を図1に示す．本データベースはファイルテーブル，フローテーブル，セッションテーブルの3つのテーブルから構成する．

ファイルテーブルには通信データを保存しているファイル全体の特徴を格納する．ファイル名， ファイルサイズ，通信パケットの総数，通信データの総量，データ通信時間等，通信ファイルの情報を把握できる．また，本

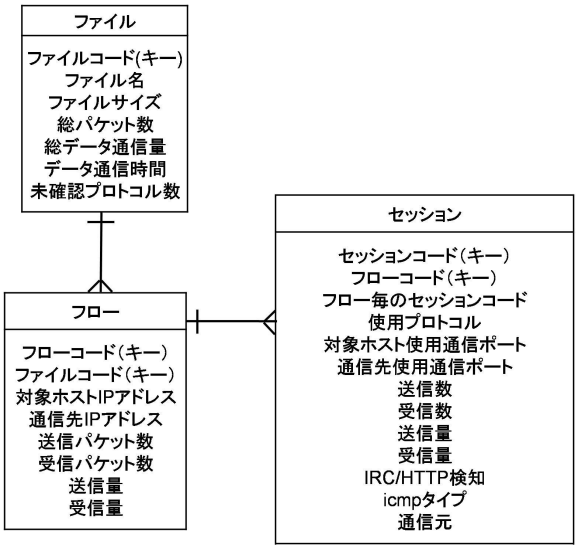

図1：データベースの構成

データベースではTCP，UDP，ICMPのプロトコルに対応しているが，上記以外のプロトコルが用いられていた場合，そのプロトコルによる通信パケット数を未確認プロトコル数として格納することで，疑わしい通信プロトコルが使用されているかどうかを判別できるようにする． その場合当該の組み合わせの通信
パケットの情報は以下のフローテーブル，セッションテーブルには格納されない．

フローテーブルには同じIPアドレスの組み合わせのIP通信フロー毎にデータが格納されている．フローテーブルに通信先ホストのIPアドレスや送受信パケット数等の特徴を格納することで，通信相手を調べることができる．

今回使用する通信データは監視対象のホストが定まっているため，監視対象のIPアドレスでない方のIPアドレスを通信先IPアドレスとして一つのカラムにまとめている．そのため，送受信は監視対象のホスト視点である．

セッションテーブルには同じフロー内で発生した複数の通信について，監視対象のホストが使用したTCP/UDP通信ポートと通信先のホストが使用した通信ポートの組み合わせを一つのセッションとみなし，セッション毎の特徴を格納している．セッションテーブルを調べることで，使用した通信ポート番号や，ポート番号に関わらずHTTPやIRCなどの特

定の通信が存在したかどうかを判別する．使用プロトコルにはTCP，UDP，ICMPのいずれかが格納され，他のプロトコルについては上記にあるようにこのテーブルでは扱われない． また，ICMPプロトコルの通信は一つの通信パケットを独立した一つのセッションとして扱う．

IRC/HTTP検知カラムでは，セッションがIRCもしくはHTTPの通信であるかどうかを確認する．セッション内の一つのパケットデータの先頭にIRC接続時に見られる命令のうちNICK，PASS，JOINコマンドのいずれかが存在した場合，そのセッションをIRCとみなし，このカラムにIRCと入力する．また，パケットデータの先頭に主なHTTP命令であるGETもしくはPOSTコマンドが存在した場合，そのセッションをHTTPとみなし，このカラムにHTTPと入力する．一般的にIRCのポートは6667， HTTPのポート番号は80であるが，これらのポート番号を用いると通信が検知や遮断されてしまう恐れがあるため，これらのTCP/UDPプロトコルを用いる際に独自のポート番号で通信を行うマルウェアが存在する．そのため，プロトコルを判定する機能を付加した．

ICMPタイプカラムでは， ICMPプロトコルの通信タイプを数値で格納する．同カラムはICMPプロトコルの場合にしか用いない．また， ICMPプロトコルの場合は両通信ポート，送受信数，送受信量のカラムは用いられない．

本データベースでは監査対象のホストと通信先のホストの通信データを格納しているが，通信を開始したのがどちらのホストであるかについてはセッションテーブルの通信元カラムに格納する．先に通信を行ったのが監査対象のホストであった場合，同カラムにsentを格納し，また先に通信を行ったのが外部のホストであった場合，同カラムにrecvを格納する．このカラムにより，セッションを開始したのがどちらのホストであるかを判別することができる．

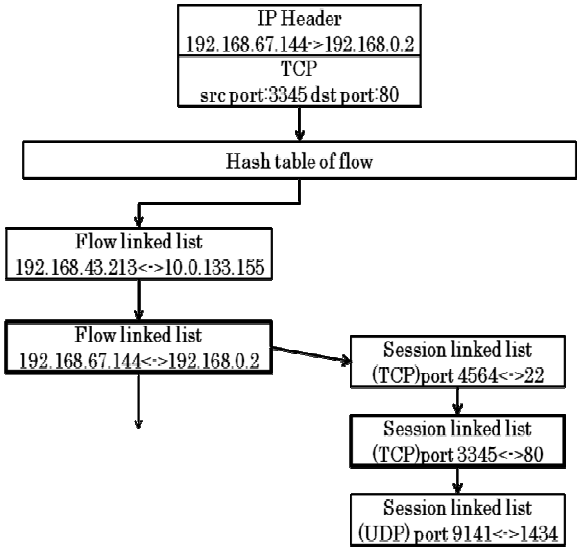

図２：解析処理の流れ

5 解析プログラムの概要

通信データを本データベース格納するため，解析プログラムを作成した．実装にはC言語を用い，OSはLinux，ライブラリはlibpcap，データベースにはmysqlを使用した．この解析プログラムは，指定した通信データファイルを開いてパケット情報からメタデータを作成し，データベースに格納する．パケット情報の解析処理の流れを図２に示す．

解析プログラムは以下の手順で処理する．

(1)データベースのフローテーブルのカラムに対応した項目を持つ構造体（以下，フロー構造体）を作成し，フロー構造体に情報を格納する．

(2)処理中のフロー検索処理を高速化するため，IPアドレスの組み合わせから作成したハッシュ値を用いたハッシュテーブルをつくり，衝突したハッシュ値のフローはリスト構造を用いて対応した．

(3)データベースのセッションテーブルのカラムに対応した項目を持つ構造体（以下，セッション構造体）を作成し，フロー構造体毎にセッション構造体のリスト構造をつくった．

(4)ハッシュテーブルに蓄積したフロー構造体を蓄積した順番にデータベースに格納する

ために，ハッシュテーブルとは別にフロー構造体の線形リスト構造を作成した．

これらの特徴を持つことで，メタデータの作成とデータベースへの格納を効率良く処理できる．

6 CCC DATAset を用いたマルウェア通信の解析

6.1 データセットの分割と解析

第5節で作成した解析プログラムを用いて，マルウェア通信データを解析した． 解析に利用したCCC DATAset 2009の通信データはWindows2000とWindows XPのホストをそれぞれ１台ずつインターネットに接続し，攻撃活動に関する通信を含んでいる．両ホストは定期的にクリーンな状態にリセットされている． 今回のデータベースは複数の通信データが存在することを前提に設計されている為，通信データを何らかの方法で分割しなければならない．一方の対象ホスト（以降，H_1）はWindows XPであり，起動時にtime.windows.comに対してNTPによる時間同期を実行する．そのため，通信データからH_1が送信元もしくは通信先ホストとなっているパケットを抽出し，前回のリセット予想時刻から一定時間以上経過した後に発生するtime.windows.comへのNTPのパケットによって通信データを分割した．これによって144個の通信データセット（以下，解析データセット）が生成された．

CCC DATAset2009を分割した解析データセットについて，解析プログラムを用いてデータベースに格納した．データベースはテーブルのカラムのレコード値を条件に検索することができる．データベースを数回検索することで得られた解析事例を記す．以下の解析結果から，本データベースを使うことによる有用性がわかる．

6.2 解析事例１:特徴的な通信の発見

セッションテーブルで通信先ホストのポート番号139番を使用したセッションを検索したところ，該当した3602個のセッションはすべて1件の通信データセットの通信であった．また，同データセットは通信先ホストのTCPポート445を使用した通信を3,599件のセッションで行っていた． これらの数値に関連性があると推測し， この通信データのセッションデータを表示したところ， 同データ内で多数の通信先ホストに対する特徴的な一連の通信を確認した．

その通信ではまず監視対象ホストがICMPプロトコルによるecho requestを4件外部ホストに送信し， その後同ホストからecho replyを受信した場合，そのホストにTCPポート139と445のパケットを送信していた． ただし，CCC DATAset の環境では内部から外部へのTCPポート139と445がフィルタリングされていたと見られ， 全ての通信先ホストから応答が観測されなかった．

この一連の特徴的な通信から， このデータセットが他のマルウェアと区別できる特徴的なマルウェアの感染活動の通信を含んでいると推測できる．

6.3 解析事例２:IRC もしくは HTTP による通信

セッションテーブルでIRC/HTTP検知カラムを検索した結果，101個のセッションがIRC通信に，87個のセッションがHTTP通信と判定された．IRC通信はいずれも独自の通信ポート番号が使用されており，64件のデータセットで確認された．HTTP通信において独自のポート番号を使用したセッションは1件しか確認されず，そのセッションを含む1件のデータセットを特定することができた．

7 考察と今後の課題

解析の結果，上記のような特筆すべき事例の他にも全てのデータセットに関する全体の傾向

として，TCP ポート 135 などの多用された通信ポートや疑わしいトランスポートプロトコルが観測されなかったことなど，解析データセットの傾向を，それぞれ数回のデータベース検索で把握することができた． このような傾向の把握が，詳細な解析の支援に役立つことが期待できる．

今回の CCC DATAset はマルウェアの検体数が限られた環境における観測であり，本稿で提案した解析手法の有用性を検証する為には，更に多くの通信データ群からメタデータの抽出を行い，実際の解析に用いることで有用性を検証する必要がある．

さらに今後は，今回データベースに実装しなかったメタデータを新たに加えデータベースを拡張することで，データベースの有用性を高めていく必要がある．今回解析したデータセットでは TCP，UDP，ICMP 以外のプロトコルによる通信は観測されなかったが，マルウェアの傾向によっては IGMP や独自のトランスポートプロトコルによる通信を行うマルウェアも多く存在する．そのため，解析の支援の為にはこれらの特徴を格納することが必要になると予想される．

参考文献

[1] 人間による Honeypot の攻撃元ログ調査を支援する User Interface の提案
http://www.iwsec.org/mws/2008/presentation.html