不正な通信の特徴を抽出した検知シグネチャ自動生成機能の設計と実装

重松 邦彦† 武田 圭史‡ 村井 純‡

†慶應義塾大学大学院 政策・メディア研究科

‡慶應義塾大学 環境情報学部
252-0882 神奈川県藤沢市遠藤 5322
sigematu@sfc.wide.ad.jp
{keiji,jun}@sfc.wide.ad.jp

あらまし 侵入検知システムにおいて不正な通信を検知するためには,その通信の特徴をシグネチャ化する必要がある. 本研究では平常時の通信と不正な通信をそれぞれファイルテーブルに保存し,フローテーブルとセッションテーブルを作成することで通信の抽象化を試みる. 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ごとのポート番号やIPアドレスや文字列などを集約し,不正な通信の特徴を定義するシグネチャを自動生成する. シグネチャは平常時の通信の誤検知を排除し,不正な通信時に出現する通信のみを検知する機能を設計,実装した.

Automated Generation of Signature for malicious detection by network traffic with specific feature

Kunihiko Shigematsu† Keiji Takeda‡ Jun Murai‡

†Faculty of Environment and Information Studies, Keio University
‡Graduate School of Media and Governance, Keio University
5322 Endo Fujisawa Kawasaki 211-8666, JAPAN
sigematu@sfc.wide.ad.jp

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

Abstract In intrusion detection system(IDS), It is necessary to detect malicious traffic, in order to generate a signature of communication data with specific feature. In this paper, we need to save normal traffic and abnormal traffic on file. And we make flow table from packet table, to attempt abstraction of abnormal traffic. Using these tables, we generate the signature of malicious traffic with specific feature automatically. We designed the function of detecting only malicious traffic.

1 はじめに

インターネットが普及し，ネットワークへの依存度が高まり，ネットワークの障害が組織に及ぼす悪影響も増大している.そのため,組織のネットワーク管理者は，ネットワークの安定性を維持するため，障害を引き起こすトラフィックを検出し，未然に障害を防止す

ることが求められる．また，障害の結果生じたトラフィックを迅速に検出し，速やかに障害に対処しなければならない．

組織内ネットワークの多くは，利用ポリシを定めており，利用ポリシに反するトラフィックを検出し，該当ユーザに警告することが組織の健全性を保つ上で重要である．そのため，多くのセキュリティ管理者は侵入検知システム(IDS)を用いてネットワークセキュリティの監視を行なっている．

既存のネットワーク型 IDS には，大きく分けて 2 種類がある．１つ目は，シグネチャ型で，ネットワーク上を流れる通信を監視し，一定のパターン（シグネチャ）との照合で不正アクセスを検出する．２つ目は，ビヘイビア型で，検査対象のプログラムを実際に動かしてその動きを監視し，不正な通信を見つけ出す手法である．これからは，シグネチャ型とビヘイビア型の併用が望ましい [1]．

本研究では，不正な通信を検知するためにネットワーク型 IDS の snort を利用する．その通信の特徴をシグネチャ化し不正な通信だけを正しく検知し，平常な通信については，検知しないシグネチャの自動生成を試みた．不正な通信で検知すべき特徴を明確にして，シグネチャを生成する．

2　問題点

本稿で取り上げる問題は，マルウェアによって発生した未整備の膨大な通信データ群の中から検知すべき通信，不正な通信のみを検知するシグネチャの生成である．今回，ネットワーク型IDSとしてSnortを用いる.以下，Snort のシグネチャについての問題点について述べる．

（1）　シグネチャの特性

Snort のシグネチャは，ステートフルな通信の記述を１つのシグネチャで記述するのは難しい．Stream を再構築する preprosessor が用意されているが負荷も実用性に乏しく実運用されていない．ステートレスな通信の記述で，検知すべき通信をメタデータから抽出し，シグネチャを生成することで解決を試みる．

（2）　FP と FN のトレードオフ

シグネチャの粒度によっては，余計なイベントまで検出してしまう FP(False Positive)と本来検出すべきなのに検出できない FN(False Negative)の問題があり，FP と FN にはトレードオフの関係がある．今回は，不正な通信からシグネチャを作ることで，FP を減らし，FN が発生しないようなシグネチャの生成をすることで解決を試みる．

3　関連研究

マルウェアのシグネチャの自動生成に関するアプローチは大きく以下の 3 種類に分類できるものと考える．

1 種類目は，パケットのヘッダ情報やペイロードの頻度，IP アドレスの散らばりを統計的に分析する手法である．EalryBird[2]では，マルウェアの不変な文字列からワームのシグネチャを自動生成するために，パケットのコンテントの頻度や散らばりを用いる．

2 種類目は，ペイロードの文字列に着目し，データマイニングを使って類似度からマルウェアの不正パケットを検出する手法である．Honeycomb[3]は，本質的に疑わしいトラフィックを集めるためにハニーポットを使い，LCS（Longest Common Subsequence）アルゴリズムを適用することによってシグネチャを生成する．また，Anomalous[4]は文字傾向の出現頻度による IDS システムで，False Positive は 1%以下という結果を出している．

3 種類目は，マルウェアの特性を抽象化し，他への感染癖や周辺アドレスへの攻撃癖などの特徴を抽象化する手法である．ヒューリスティックビヘイビア型の検知手法を用いた

AntiBot[5]などがある.

4 事前準備

4.1 概要

シグネチャの生成には,事前準備が必要となる．まず，4-2 で，不正通信の作成方法を説明し，4-3 で，検知すべき通信の抽象化を試みる．抽象化することで，検知すべき通信を明確にする．4-4 で，作成した DB から不正通信で検知すべき通信を特定する．また，4-5 で，検知すべき通信からシグネチャを作成するまでのフローについて説明する．

4.2 不正な通信の作成

Malware Domain List[6] では悪意のある Web サイトの URL をまとめており，攻撃コードが含まれる Web サイトも公開されている．この一覧に含まれる URL を 2008 年 7 月 16 日から同年 10 月 2 日までの間に仮想マシン上で動作する Internet Explorer(Version: 6.0.2600.0000) によって閲覧し，20 分間通信を監視することにより，180 件の URL を調査した.

4.3 検知すべき通信を抽出する DB

特定の通信データを抽出するためには，全ての通信データを保存する必要がある．検知すべき通信を把握するために，各通信の通信情報を持つ DB を作成する．これは,収集した通信データを保存した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に対して，情報の抽出に利用する[7].

まず，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出来る．また，確認プロトコルとして TCP, UDP, ICMP 以外を未確認プロトコルとした.

フローテーブルには,同じ IP アドレスの組み合わせから IP 通信フロー毎にデータが格納されている．フローテーブルに送信元 IP アドレス,送信先 IP アドレスや送受信パケット数送受量を格納する．フローの特徴を格納することで，通信相手と通信相手とのパケット数とパケット量を求めることが出来る.

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.使用プロトコルは,TCP,UDP,ICMP のいずれかが格納される．また，ICMP は一つの通信パケットとして独立したものとして扱う.

ファイルコード（キー）
ファイル名
ファイルサイズ
開始時間
終了時間
総パケット数
総データ通信量
データ通信時間
未確認プロトコル

図 1：ファイルテーブル

フローコード（キー）
ファイルコード（キー）
送信元 IP アドレス(固定値)
送信先 IP アドレス
送信パケット数
受信パケット数
送信量
受信量

図 2:フローテーブル

セッションコード（キー）
フローコード（キー）

フロー毎のセッションコード
使用プロトコル
対象ホスト使用通信ポート
通信先使用通信ポート
送信数
受信数
受信量
受信量
IRC/HTTP 検知
ICMP タイプ

図 3：セッションテーブル

4.4　解析結果

図 3 にあるセッションテーブルで不正な通信で利用される特徴的な通信先使用ポートを確認したところ，特定のポート番号が多く利用されていることが分かった．図 4.1 にあるように不正な通信の生成には，Web サイト閲覧による通信の取得にも関わらず，全セッションが 68665 中，ポート番号 25，110，135 が多く現れた．これは，恐らく不正通信の中にマルウェアに感染後の行動に利用されたと考えられる．

ポート番号	出現回数
25	4275
80	37354
110	829
135	2011
443	18203
その他	5993

図 4:不正通信で利用されたポート番号

実際，フローテーブルで，ポート 25 を利用するまでの処理を確認すると，HTTP でサイトへアクセス後，DNS へ問い合わせしポート 25 を利用してメールを送信していることが確認出来た．これらの通信の特徴からマルウェアをダウンロードし，SPAM メールを送信するマルウェアであると考えられる．

4.5　生成するシグネチャのフロー

シグネチャ化の対象とする通信は，HTTP でファイルをダウンロード後，他のホストへ通信を行なっているものをマルウェアに感染と定義する．また，他のホストへ通信を行なっていないものを感染失敗と捉え，シグネチャ化はしない．以下，図 6 でシグネチャ生成フローを説明する．

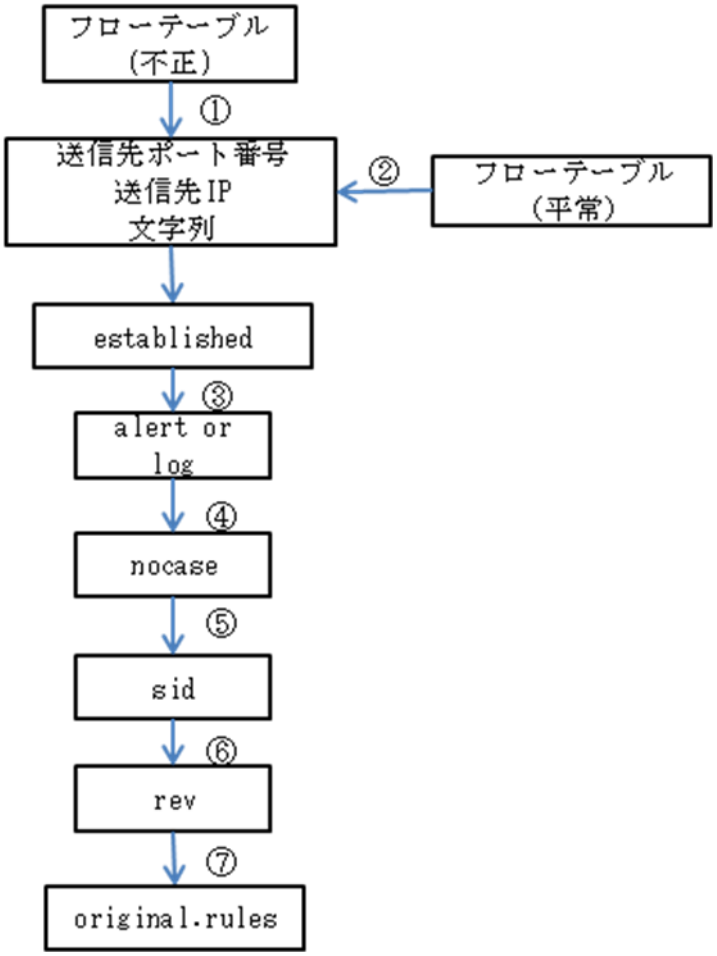

図 5:シグネチャ生成フロー

① フローテーブルから送信先IPアドレスと送信先ポート番号と文字列を抽出する．　58.65.239.115→
58.65.239.0/24
文字列については，HTTPによるGetでexeファイルの名前を取得することに

した.
(例GET/gh3ghwd/ns84.exe→ns84.exe

② フローテーブル(不正な通信)とフローテーブル(平常な通信)通信を比較し，不正な通信の中に平常な通信が含まれていないかチェックする.

③ 不正な通信の中に，平常な通信と同等のパケットがあった場合は，logを出力し，不正な通信のみに現れるパケット場合は，alertとする.

④ nocaseは，大文字小文字を区別しない.固定文字とする.

⑤ sidは，ユニークに番号を割り当てる.

⑥ revは，1を割り当てる．固定文字とする.

⑦ original.rulesファイルに①～⑥の情報を書き込む.

5 実験結果

5.1 シグネチャの例

実際に，自動で作成されたシグネチャのサンプルを紹介する．ここで作成したシグネチャは，図5のフローを通して，自動で作成したものである.

```
alert tcp any any -> 58.65.239.115 80
(msg:"original GET ns84.exe ";
flow:established,to_server;
nocase;content:" ns84.exe ";sid:10000000;
rev:1;)
```

図6:シグネチャの生成例

5.2 評価

評価は，Snort の最新のシグネチャをダウンロードした状態とシグネチャ生成フローによって生成されたシグネチャを使って比較した．項目は，TP（True Positive）と FP と FN を利用する.

今回，検知対象の通信は，全部で 53 件存在する．その定義は，4.5 生成するシグネチャのフローでも述べた通り，マルウェアに感染後，他のホストへ通信しているものをシグネチャの生成対象とした.

また，不正な通信で作成されたシグネチャが，平常時の通信で，検知しないことも確認する．ただし，平常時の通信と不正な通信では共通する通信については，log として出力されるのが正しい.

全部で，53 件の攻撃通信があるため，生成したシグネチャで 53 件検知するのが望ましく，平常時の通信では，0 件の検知であることが望ましい．以下に今回の実験結果を説明する.

Snort のデフォルトのシグネチャでは，不正な通信を 2 件しか検知しなかった．そのうち 2 件は， SMTP で，マルウェアに感染した仮想マシン上のクライアントから，他のホストへ SPAM メールを送る通信だった.

また，FP として検知したものは，25 件あった．そのうち 13 件は，外部のマシンから ICMP が送られてきたものだった．残りは，SNMP リクエストなど多岐に渡る．53 件の検知を期待していたにも関わらず，TP が 2 件しか無く，残りの 51 件は FN と言うことが出来る.

生成シグネチャを試験する際は，Snort デフォルトのシグネチャを削除し実験を行なった．その結果，検知して欲しい 53 件を検知することが出来た．また，FP と FN は発生せず，生成したシグネチャで，53 件全てのシグネチャで検知出来た.

	TP	FP	FN
Snort デフォルト	2	25	51
生成シグネチャ	53	0	0

図 8:不正な通信

	TP	FP	FN
Snort デフォルト	-	18	-
生成シグネチャ	0	0	-

図 9:平常な通信

5.3 結論

本稿では，不正通信から作成したシグネチャを作成することで，平常時の通信で誤検知せず，不正な通信時に出現する通信のみ検知することに成功した．これにより，不正通信を特定出来た場合に，正しくシグネチャを作成し，検知することが出来た．また，平常時の通信では，作成されたシグネチャでは検知しなかった．

以上から,不正な通信を抽象化する DB を用いて，シグネチャ化する本手法は，検知すべき通信の抽出に成功し，TP の向上と，FP の低減に成功した．

6 今後の課題

限られた通信との比較では，IP アドレスや文字列など不正な通信となる情報は固定されており，まだ実用的とは言えない．例えば，文字列の抽出でも本手法では，GET で取得してくる文字列を対象としている．今回は不正通信を学習データとして，シグネチャを作成したがまだ実用化までは至っていない．シグネチャを抽象化して検知するアルゴリズムについての考察を進める必要がある．

7 参考文献

[1]ITMedia エンタープライズ: “ビヘイビア法に注目」，IPA が未知のウイルス検出技術に関する報告書”http://www.itmedia.co.jp/enterprise/0404/23/epn02.html(2004)

[2] Singh, S., Eitan, C., Varghese, G., Savage, S.：Automated worm fingerprinting. In: 6th Symposium on Operating Systems Design and Implementation (OSDI), December (2004)

[3] Kreibich, C., Crowcroft, J.: Honeycomb: creating intrusion detection signatures using honeypots. SIGCOMM Comput. Commun. Rev. 34(1), 51□56 (2004)

[4] Christodorescu, M., Jha, S., Seshia, S., Song, D., Bryant, R.: Semantics-aware malware detection. In: Proceedings of the IEEE Symposium on Security and Privacy(2005)

[5]Symantec.Norton.antibot. http://www.symantec.com/norton/antibot, 2008.

[6] MDL. Malware Domain List. http://www.malwaredomainlist.com/

[7] 榊 辰哉 水谷 正慶 武田圭史 村井純″メタデータ作成によるマルウェア通信解析手法の提案″ MWS2009