IDSと連携した高速に伝播するワームのシグネチャ自動生成機構の設計と実装

金井 瑛† 水谷 正慶† 白畑 真‡ 南 政樹‡ 村井 純†‡

慶應義塾大学環境情報学部† 慶應義塾大学大学院政策・メディア研究科‡

ワームによるインターネット上の被害は増大している．その対抗策としてパケットを監視するIDSが挙げられる．ミスユース型IDSは攻撃のシグネチャをあらかじめ定義しなければならないため，既知の攻撃に対しては有効であるが未知の攻撃に対しては効果が期待できない．これを解決するために自動シグネチャ生成機構の研究が行われているが，IDSが既知であるシグネチャを重複して作成する問題がある．本稿では，IDSでは対応できない高速に伝播する未知のワームのシグネチャを自動的かつ既知のIDSシグネチャと重複無く生成する手法を提案し，設計，実装した．評価の結果，本機構によりIDSシグネチャと重複無く，高い負荷のネットワークでも正常にワームのシグネチャが自動生成されることが示された．

The design and implimentation of Automated Worm Fingerprinting for high speed transmission unknown worms with IDS

Akira Kanai†, Masayoshi Mizutani†, Shin Shirahata‡, Masaki Minami‡, Jun Murai†

†Faculty of Environmental Information, Keio University

‡Graduate School of Media and Governance, Keio University

Damages from worm attacks are increasing on the Internet. IDS is a system which observe packets to detect attacks. IDS has to define attacking signature in advance. Therefore it is effective in defined attacks, however it is not effective in unknown attacks. Automated Worm Fingerprinting is well researched to solve this problem, however it creates duplicated signatures. This research suggest and design Automated Worm Fingerprinting without duplication method for high speed transmission unknown worms which IDS is not supporting. In this research, we also implements a prototype system to evaluate. From the result of evaluation, this system indicates it can automatically create worms fingerprint without duplication in high load network.

1 背景

近年，インターネット上のワームやコンピュータウィルスによる被害は増大し，インターネットに接続されたホストあるいはネットワークそのものが脅かされている．2003年に流行したSlammerワーム[1]は10分に満たない時間で脆弱性を持ったホストの99％以上に感染した[2]．一度流行が始まったワームの感染拡大を防ぐのは極めて困難であり，高い感染力を持ち高速に伝播するワームは数分以内，あるいはより早い時間でなんらかの対応をしなければ24時間以内に根絶させることは困難との報告がある[3]．よって，高速に伝播するワームの感染拡大を防ぐには，極めて迅速な対応が必須である．

ワームや悪意のある攻撃を監視する手法の代表的な手法として，ミスユース型IDS(Intrusion Detection System)が挙げられる．IDSはネットワークのトラフィックを監視し，プロトコル番号や宛先ポート番号，ペイロードの内容が既知の攻撃に固有のパターン（シグネチャ）を含むか否かにより攻撃を検知する．

しかし，従来のミスユース型IDSでは高速に伝播するワームの検知は困難である．シグネチャは主にIDSのベンダやボランティアによって作成されるが，新しいワームを発見，解析し，シグネチャを作成する作業は時間がかかり，その間新しいワームを検知できない．

故にミスユース型IDSは最新の高速に伝播するワームの検知に対しては効果が期待できない．

2 自動シグネチャ生成機構とその問題点

ミスユース型IDSが既知の攻撃にしか有効ではない問題を解決するために，EarlyBird[4]などの自動シグネチャ生成機構が研究されている．これらの機

[1],[2]Keio University Shonan Fujisawa Campus
5322, Endo, Fujisawa, Kanagawa 252, Japan
E-Mail: kanai@sfc.wide.ad.jp

構は定常的にネットワークのトラフィックを観測し，ポート毎のパケット流量や，パケットの特徴を元にトラフィックの異常を検知する．そして，ワームが発生したと判断されればそれをワームのシグネチャとして出力する機能を持つ．

しかし，これらの機構はIDSとは独立したシステムであり，既知の攻撃に対するシグネチャを生成する可能性がある．IDSは正しく攻撃を検知できるシグネチャが1つ登録されていればその攻撃を検知できる．そのため，ある攻撃を示すシグネチャが複数存在していると1つのパケットに対して冗長なマッチングを行うこととなる．この結果，IDSやインライン型のIDSであるミスユース型IPS(Intrusion Prevention System)のスループット低下を引き起こす可能性がある．

更に1つの攻撃に対して複数回の検知ログが残る．冗長なログは，ログによる解析やログ整理の際に混乱の元となりうる．

自動シグネチャ生成機構を用いることによりミスユース型IDSで高速に伝播するワームの検知が可能になるが，既存の機構では重複したシグネチャが生成されてしまう問題がある．

3 重複したシグネチャ生成を回避する手法

本研究では，既存の自動シグネチャ生成機構が，既知の攻撃を示すシグネチャも生成してしまう問題点を解決するために，IDSと連携した自動シグネチャ生成手法を提案する．

既存の研究が重複したシグネチャを生成する原因として，自動シグネチャ生成機構とIDSが独立して動作していることが挙げられる．

重複したシグネチャが生成される問題を解決する手法として，自動シグネチャ生成機構がIDSのルールセットを読み込むという方法が考えられる．しかし，パケットとルールセットの比較は重複してIDSの処理を行うため，非効率である．

そこで本稿では，観測するトラフィックをIDSで走査した後に，自動シグネチャ生成機構に渡す手法を提案する．本手法では，IPSで正常と判断されたパケットのみを自動シグネチャ生成機構で処理する．IPSによって既知の攻撃はフィルタリングされ，自動シグネチャ生成機構にそのパケットが渡ることはない．これによって既知の攻撃を示すシグネチャが生成されることを抑制できる．また，IPSは自動シグネチャ生成機構と独立して動作しているために，IPSのシグネチャフォーマットに影響されることなく本手法を適応できる．

本手法の概要を図1に示す．

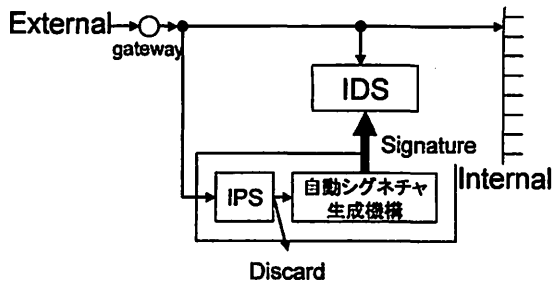

図1: 提案手法

本手法により，既知の攻撃シグネチャを生成されることはなくなり，また，冗長なログが生成されることもなくなる．

4 要件

4.1 即時性

ここで述べる即時性には2つの側面がある．1つはルール生成，もう1つはルール反映である．

ルール生成の即時性とは，ワームの伝播を検知すると同時にワームのシグネチャを生成することである．ルール生成の即時性は第1節で述べたとおり，近年のワームには可能な限り早い対応が求められているために，高速に伝播するワームの流行をIDSで検知することが必須である．

ルール反映の即時性とは，シグネチャが生成されると同時にIDSのルールセットとして反映させることである．自動的に作成されたシグネチャはそのままIDSのシグネチャとして適応させることが出来ない．IDSの実装によってシグネチャの書式は異なるために，自動的に作成したシグネチャをIDSに適応した書式に変換する必要がある．

4.2 バーストトラフィックへの対応

高速に伝播するワームが流行するとネットワークのトラフィックが増大する．トラフィックが一定以上に増加すると，カーネルでの処理や回線の帯域幅に起因するパケットの取りこぼしが発生する．管理しているネットワーク全体のトラフィックをIDSで観測する場合，外部のネットワークとの境界で観測を行うことが一般的である．ワームの感染規模によっては、数Mpps以上のワームトラフィックが発生する場合ある．そのため，高速に伝播するワームの発生時にはパケットの取りこぼしの発生が予想される．

本機構はトラフィックの観測において，パケットの取りこぼしが発生した場合であっても取りこぼしていないときと同様のシグネチャを生成しなければならない．

5 設計

第4節で述べた要件を踏まえ，自動シグネチャ生成機構の設計を行った．本機構は大きく二つのモジュールで構成できる．モジュール間の関係を図2に示し，以下で各モジュールについて述べる．

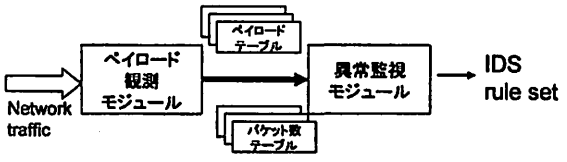

図 2: 各モジュールの関係

5.1 ペイロード観測モジュール

ペイロード観測モジュールは，パケットのペイロード部分を保持、比較するモジュールである．本モジュールでは入力されたパケットを走査し，TCP あるいは UDP のポート番号とペイロードのパターンを抽出する．また，ポート別にパターンをカウントし，さらにパターンの数をあらかじめ定めた単位時間毎に記録する．

本稿ではネットワーク上を流れるある2つのパケットにおいて共通するペイロード部分をパターンと定義する．本モジュールはパケットが入力されると，そのペイロードをポート番号毎のリスト (ペイロードリスト) に追加し，さらにポート番号毎のパケット数をカウントする．次にペイロードリストと別に作成されたパターンリストに一致するパターンがあるかを調べる．新しいパケットが到着すると，ペイロードをペイロードリストのデータと比較し，同じパターンが見つかればそれをポート毎のリスト (パターンリスト) に追加するか，既に追加されているパターンであれば，カウントする．

図 3に HTTP の Well-known ポートである TCP のポート 80 番を例にとって，本モジュールの動作を示す．まず，最初に届いたパケットは比較するペイロードが存在しないため，そのままペイロードリストに追加される (Payload1)．次に届いたパケットは新たなペイロードの情報としてペイロードリストに追加され (Payload2)，ペイロードリストに含まれるデータ (Payload1) と比較する．例えば HTTP の基本的な構文である，「GET」が両方のパケットに含まれていれば，その GET を 80 番ポートのパターンリストに追加する (Pattern1)．更にパケットが到達した場合，同様にペイロード (Payload3) はペイロードリストのデータ (Payload1,2) と比較され，パターンリストのデータ (Pattern1,2) とも比較される．

図 4にポート毎のパケット数と，ポート毎のパターン数を保持するテーブルについて示す．

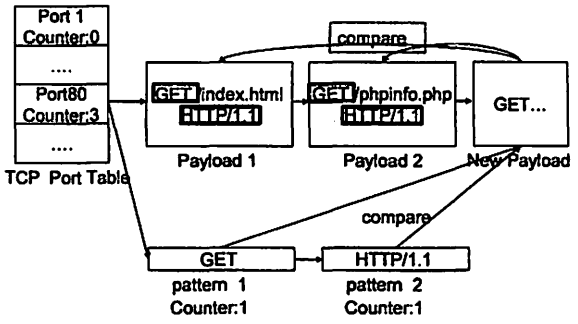

図 3: パターンの抽出

パケット数テーブルは，ポート毎のパケット数をカウントするテーブルであり，IP プロトコル番号とポート番号を元に単位時間当たりのパケット数を保持する．

パターンテーブルは，ポート毎のパターンとその数をカウントするテーブルであり，IP プロトコル番号，ポート番号および，検知したパターンの文字列と，単位時間当たりに発生したパターン数を保持する．

protocol	port	count
6	25	213
6	80	6143
.....	....	...
17	1434	2134

パケット数テーブル

protocol	port	pattern	count
6	25	HELO	13
6	80	GET ~/\n...	234
6	80	<HTML>¥n	161
.....			.....

パターンテーブル

図 4: パケット数とパターン数のカウント

一定時間毎に，本モジュールは測定開始時刻と各テーブルを異常監視モジュールに伝達し，各テーブルのカウンタをリセットする．

5.2 異常監視モジュール

本モジュールでは，ペイロード観測モジュールから伝達されたポート毎のパケット数とパターン数の推移からワームの発生を検知し，ワームのシグネチャを生成する．多くのワームは特定のサービスを対象とするので，そのようなワームが流行すると特定のサービスポートに対するアクセスが急激に増加する．この特性を利用して定常的にトラフィックを観測し，異常を検知したらワームのシグネチャを生成する必要がある．

まず，ペイロード観測モジュールで作成した過去のパケット数に注目する．ポート毎のパケット数に対してあらかじめ閾値を定めておき，その閾値を越

えていたらワーム発生の可能性があると判断して，パターン数の推移に注目する．パケット数があらかじめ設定してある以上に増加した場合，それをワームのシグネチャと判断する．

あるパターンを含むパケットがワームとして判断されると，そのパターンをペイロードに含むパケットに対するシグネチャを生成する．

5.3 IDSとの連携

ワームと判断したパターンをIDSに反映させるためにIDSとの連携が必要となる．

異常監視モジュールから得られる情報は，プロトコル番号と宛先ポート番号，ペイロードの情報が図4のようにデータが蓄積されている．異常監視モジュールからシグネチャ生成の要求が発生した際にはこれらの情報を用いて，IDSに適したシグネチャを生成する．その後，IDSのルールセットにシグネチャを追加し，IDSに変更を反映させる処理を行う．

6 実装

第5節の設計に基づいてDebian GNU/Linux 3.1 (Kernel 2.6.8) 上でプロトタイプ実装を行った．ペイロード観測モジュールの実装にはC言語を用いて，パケット取得ライブラリとしてLibnids[6]を利用した．異常監視モジュールの実装とIDSの連携はPHP4.3とzshスクリプトを用いた．本機構は約1500行のC言語とPHPのソースコードおよびzshスクリプトで実装した．

6.1 IDSとの連携の実装

本プロトタイプではIDSとの連携のためにLinuxで広く使われているパケットフィルタリングツールのiptablesを用いた．iptablesとの連携を実現するためのIPSとしてSnort[7]の機能であるSnort-inline[8]を使用した．iptablesとSnortおよび本機構の関係を図5に示す．まず，入力されてきたパケットはiptables

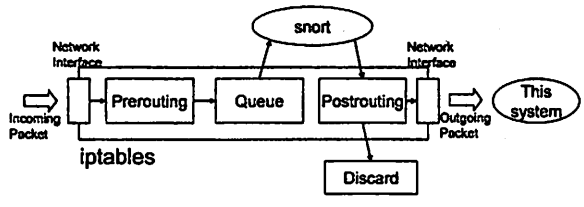

図 5: 本機構の実装概要

によってSnortに転送される．パケットの検査が終わると再びパケットはiptablesに再度処理され，パケットが攻撃と判断された場合，パケットを破棄し，攻撃でないと判断されれば，指定した出力インタフェースにデータを転送する．本機構は出力インタフェースからのデータを受け取り，処理をする．

7 評価

本節では本機構が既存の問題を解決し，実際に要件を満たしているかを評価する．

7.1 実験環境

第5節の設計で述べたとおり，本機構はペイロード観測モジュールが異常監視モジュールにデータを伝達する単位時間及びポート毎のパケットの推移がワームであるかを判断する条件をあらかじめ設定する．

異常監視モジュールに対するデータの伝達間隔については，ワームが発生してから1分以内にワームの遮断などを行えば，24時間以内の根絶が見込める[3]ことから30秒と設定した．

また，ワームであるかの判断は，実験に用いるネットワークでは通常観測されない単位時間あたりに1500カウント以上の同一パターンとした．

図6に示す3台のホストを用いて評価を行った．ホストAとホストBでは共に第6節でプロトタイプ実装した本機構が動作している．ホストAのSnortにはSlammerのシグネチャが登録されており，ホストBのSnortからはSlammerのシグネチャを削除してある．

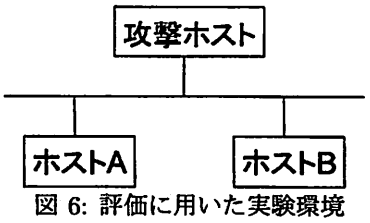

図 6: 評価に用いた実験環境

7.2 重複しないシグネチャの生成

本機構により，IPSが既知であるワームのシグネチャが生成されないことを示す．

攻撃ホストからにホストAとホストBに対してSlammerワームを毎秒100件送信した．

実験の結果，ホストAではSlammerワームはIPSで検知され，パケットが破棄されるために本機構ではシグネチャは生成されなかった．

対して，ホストBではSlammerワームはIPSを通過し，そのまま出力インタフェースで本機構によって処理され，Slammerワームに対するシグネチャが新たに追加された．

7.3 処理能力と検知能力の評価

本機構はインタフェースに入力されたトラフィックをすべて処理するが，特にワームの発生時は第4.2節

表 1: パケットロス率とワーム検知

パケット数 (pps)	パケットロス率 (%)	生成
10	0	×
60	0	○
60	19.4	×
100	0	○
100	21.2	○

で述べたとおり，パケットの取りこぼしの発生が考えられる．

本評価では，パケットロスが発生する環境においても，ワームの自動生成が行われることを確認する．攻撃ホストからホストBに対してSlammerワームと悪意のないパケットを同時に流す．悪意の無いパケットの流量を増加させることによりホストBでのパケットロス率を増加させる．これによりバーストトラフィックへの対応を評価できる．

実験の結果を表1に示す．この結果から，通常のパケットが多く流れるネットワークにおいて，ワームのパケットが，本機構がワームと判断する閾値前後の流量で流れてくると，パケットロスによって観測できるワームのパケット数が制限され，正常に攻撃を検知できない可能性があることが分かった．

しかし，十分なワームの流量が確認できる状態においては，同等程度のパケットロスでもシグネチャを生成できた．Slammerのケースではノードあたり秒間100パケット以上のワームが送信されたという報告があり[2]，高速に伝播するワームは今回用いた閾値よりも大きな規模で流行すると考えられる．故に，ワームによってパケットロスが発生するようなトラフィック状況においても，本機構はワームのシグネチャを有効に自動生成できる．

8 今後の課題

ネットワークの規模や目的によって，トラフィックの量やその性質は異なるため，定常時のネットワーク流量を流量も同様に異なる．

また，多くのワームは攻撃先のアドレスを特定しないため，ネットワークの規模が大きいほど，ネットワークに入ってくるワームの量は多く，異常時の単位時間当たりのパケット数の判定は困難である．

このように，定常時と異常時の判断は容易ではないため，閾値による判断は，複数のネットワークで利用する上で大きな課題となる．よって，定常時のデータから異常を自動的に検知するような改善が必要である．

9 おわりに

本稿では，ミスユース型IDSでは対応できない高速に伝播する未知のワームのシグネチャを自動的かつ既知のIDSシグネチャと重複なく生成する手法を提案し，設計，実装した．

評価の結果，本機構によりIDSシグネチャと重複無く，高負荷下のネットワークにおいても正常にワームのシグネチャが自動生成されることが示された．

参考文献

[1] *CERT Advisory CA-2003-04 MS-SQL Server Worm*
http://www.cert.org/advisories/CA-2003-04.html

[2] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford and Nicholas Weaver, *"The Spread of the Sapphire/Slammer Worm"*, CAIDA Technical Report 2003.
http://www.cs.berkeley.edu/ nweaver/sapphire/

[3] David Moore, Colleen Shannon, Geoffrey M. Voelker and Stefan Savage, *"Internet quarantine: Requirements for containing self-propagating code"*, in INFOCOM 2003.
http://citeseer.ist.psu.edu/moore03internet.html

[4] Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, *"The EarlyBird System for Real-time Detection of Unknown Worms"*, Technical Report CS2003-0761, August 2003.
http://www.cs.ucsd.edu/Dienst/UI/2.0/Describe/ncstrl.ucsd_cse/CS2003-0761

[5] *Libnet*, George Foot
http://libnet.sourceforge.net/

[6] *Libnids*, Rafal Wojtczuk
http://libnids.sourceforge.net/

[7] *Snort*, Martin Roesch, "SNORT-LIGHTWEIGHT INTRUSION DETECTION FOR NETWORKS", USENIX LISA '99 Conference, 1999.
http://www.snort.org/

[8] *Snort-Inline*
http://snort-inline.sourceforge.net/