社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2005－ＣＳＥＣ－30（１）
2005／7／21

囮サーバで送受信されたパケット系列を統計分析することによるワーム検知システムの提案

片岡　真紀 †　石毛　由美子†　萬谷　暢崇‡　大橋　史治‡

†株式会社 NTT データ 公共ビジネス事業本部 ナショナルセキュリティ BU
‡警察庁 情報通信局 情報技術解析課, 〒100-8947　東京都千代田区霞ヶ関 2-1-2
E-mail: kataokamk@nttdata.co.jp, ishigeym@nttdatacs.co.jp, ‡{nmantani04, foohashi04}@npa.go.jp

あらまし　インターネット上の囮サーバで送受信されたパケットデータを統計分析することによって、未知のワーム及びワームの亜種を検知するシステムを提案する。提案システムでは、囮サーバに対する不正アクセスを、複数のコネクションを含むパケットの一連の流れであるパケットフローとして表現し、ベクトル空間モデルや編集距離を用いることでパケットフロー間の類似度を計算する。これにより、囮サーバに対する不正アクセスの種類ごとの発生頻度や他の不正アクセスとの類似性を知ることができるので、未知のワームの発見やワームの亜種の発見が可能となる。

キーワード　不正アクセス、ワーム検知、ベクトル空間モデル、編集距離

A Proposal of Worm Detection System by Taking a Statistics of Series of Packets to Honeypot

Maki KATAOKA† Yumiko ISHIGE† Nobutaka MANTANI‡ and Humiharu OHASHI‡

†National Security Business Unit, Public Business Sector, NTT Data Corporation,
‡High-Tech Crime Technology Division, Information Communications Bureau, National Police Agency, 2-1-2, Kasumigaseki, Chiyoda, Tokyo 100-8947
E-mail: kataokamk@nttdata.co.jp, ishigeym@nttdatacs.co.jp, ‡{nmantani04, foohashi04}@npa.go.jp

Abstract　A worm detection system tells us when an unknown worm occurs as well as its attributes. In the proposed system, traffic from the Internet to the honeypot is divided into a series of packets, which include some established TCP connections. The similarity between one series of packets and another are calculated by using the vector space model and the edit distance. As a result, we were able to know that how many times each incident happened and how similar each incident was to others incident. Finally, we were able to find some unknown worms and their attributes.

Keyword　Computer Security Incident, Worm Detection, Vector Space Model, Edit Distance

1. はじめに

近年、セキュリティホールが発見されてから、そのセキュリティホールを悪用して感染するワームが出現するまでの期間が短くなってきている（表 1）。そのため、ワームの感染が拡大するまでの間にパッチの適用やシグネチャ及びウィルス定義ファイルのアップデートなどの対策を行うといった従来型のワーム対策では、不十分になってきている。そこで、今後のワーム対策技術として、未知のワームを検知する機能が重要になってくると考えられる。

表 1 新種のワームが出現するまでの期間

発生時期	名称	パッチ公開からワーム出現までの期間
2001 年	Nimda	120 日
2003 年	SQLSlammer	185 日
	MSBlaster	26 日
2004 年	Sasser	17 日

2. 既存の未知ワーム検知技術の問題点

ワーム検知の方法としては、RealSecure Network Sensor のような侵入検知製品を使用する方法や Norton AntiVirus のようなアンチウィルスソフトを使用する方法がある。しかし、どちらの方法についても、あらかじめウィルスに関して情報を記述する必要があり、未知のウィルスに対応できないという問題がある。

そこで、近年、未知ワームの検知方法について、非常に多くの研究がなされている（文献[1]）。例えば、山西らによる研究（文献[2]）では、正常状態のネットワークトラヒックの定義をおこない、正常との違いを監視することで、ワームだけでなく、ネットワーク侵入やなりすまし、コンピュータ障害を検知している。また、Dagon らによる研究（文献[3]）では、囮サーバのメモリー状態、パケット送信状態、ディスク状態を監視することでワームを検知している。しかし、どちらの手法についても、通常とは異なる事象がおきていることは検知できるが、その事象の原因の特定ができないという問題が残る。

3. ワームの分類と特徴

本章では、ワームの分類と特徴について確認する。ワームは「寄生先を必要としない独立したプログラムで、自己伝染機能を持つもの」と定義されている。そして、セキュリティホールを狙って感染する「セキュリティホール悪用型ワーム」とメールを感染媒体とする「メール感染型ワーム」の 2 種類に大別できる。このうち、「セキュリティホール悪用型ワーム」では、特にユーザの操作を必要とせず、自動的に感染を拡大するため、セキュリティホールが発見されてからそのセキュリティホールを悪用して感染するワームが出現するまでの期間が短期化すると、感染件数が急増しやすいといえる。例えば、「セキュリティホール悪用型ワーム」の 1 つである SQLSlammer は、全世界の脆弱性のあるホストの 90%に 10 分以内で感染したといわれている。

次に、「セキュリティホール悪用型ワーム」の特徴を以下に示す。

- ネットワーク経由でコンピュータのセキュリティホールを直接狙って侵入し、感染する。
- 感染活動のため、ランダムな IP アドレスに対して、多量の攻撃パケットを送信するため、インターネット上で同一の特徴をもつワームを多数確認することができる。
- 感染活動において、セキュリティホールの攻撃、バックドア作成、スクリプト送信、本体の送信等のため、複数回の TCP コネクション確立を行うものが多い（MSBlaster、Sasser など）。
- 感染活動において TCP 接続を利用する場合、IP アドレス詐称ができないので、IP アドレス詐称を行っていないものが多いと考えられる。（ただし、この場合の攻撃者は、すでにそのワームに感染した被害者である可能性が高い。）

そこで、この「セキュリティホール悪用型ワーム」の性質を利用して、未知のワーム及びワームの亜種を検知するシステムの提案を行う。

4. 提案システム

提案システムでは、インターネット上の囮サーバで送受信されたパケットデータを統計分析することによって、未知のワーム及びワームの亜種を検知する。提案システムの特徴を以下に示す。

- パケットやコネクション単位ではなく、複数コネクションを含むパケットフロー単位で分析を行う（4.2参照）。
- パケットフローを 3 つの部分に分解して、文書の検索において文書の類似度を計算するために使用される手法やエラーを許す文字列照合の手法を複合して、類似度計算に適用している（4.3参照）。
- ユーザがワーム名称を登録することにより、検知した未知のワームにおいて、既存のワームの亜種に属するか否かを判別することができる（4.4参照）。

4.1. 提案システムの流れ

提案システムの流れを以下に示す。

(1) 囮サーバをインターネットに接続して取得したダンプデータからパケットフローの作成を行う。
(2) パケットフロー間の類似度を計算する。
(3) 類似度の分布表を作成する。
(4) 利用者は、類似度の分布表において、必要に応じてパケットフローの名称を登録する。

4.2. パケットフローの作成

多くのワーム検知製品は、パケットやコネクションごとに、あらかじめ登録された攻撃手法のパターンとマッチングさせることにより、ワームを検知する。しかし、最近のワームは、感染活動において、セキュリティホールの攻撃、バックドア作成、スクリプト送信、ワーム本体の送信等のため複数回の TCP コネクション確立を行うものが多い。そこで、提案システムでは、パケットやコネクション単位ではなく、複数コネクションを含むパケットフロー単位で分析を行う。なお、このパケットフローは、ダンプデータを送信元 IP アドレスごとに分類することにより作成する。パケットフローの例を図 1に示す。

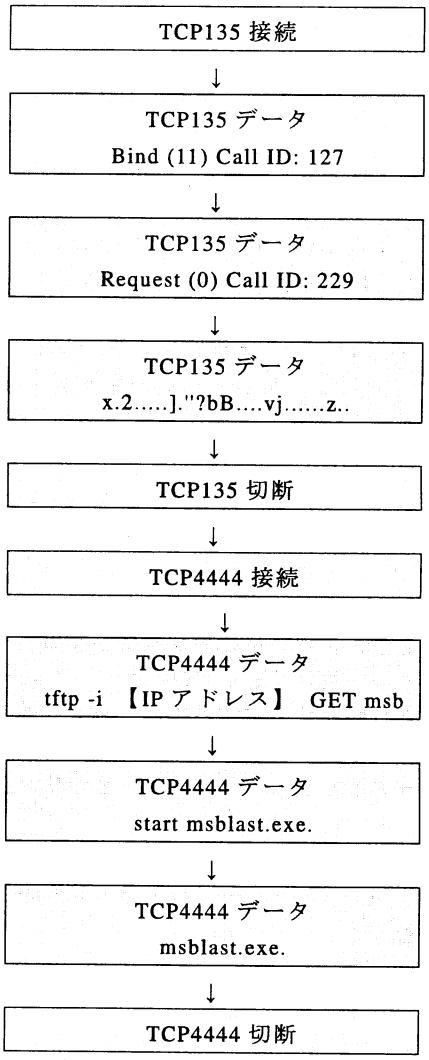

図 1 パケットフローの例

4.3. 類似度計算の手法

パケットフローは、非常に多くの情報を含むため、そのままの状態で類似度を計算するのは難しい。そこで、提案システムでは、パケットフローを詳細プロトコル情報、データ情報、パケット並び順情報の 3 つの部分に分割して、類似度計算を行う。分割方法を以下に示す。

【詳細プロトコル情報】TCP、UDP より上位の HTTP、FTP、DCERPC 等のデータ（図 1における「Bind (11) Call ID: 127」「Request (0) Call ID: 229」）

【データ情報】詳細なプロトコル情報が解釈できないデータ（図 1における「x.2.....].''?bB....vj.....z..」「tftp -i 【IP アドレス】 GET msb」「start msblast.exe.」「msblast.exe.」）

【パケットの並び順】パケットフローの各パケットについて、プロトコルの種類（ICMP、TCP、UDP）、送信先ポート番号、コネクション状態（接続、データ、切断、リセット）を取り出し、時系列に並べたもの（図 1における「TCP135 接続→TCP135 データ→TCP135 データ→TCP135 データ→TCP135 切断→TCP4444 接続→TCP4444 データ→TCP4444 データ→TCP4444 データ→TCP4444 切断」）

分割した 3 つの部分のそれぞれの類似度の計算

は、文書の検索において文書の類似度を計算するために使用される手法である「ベクトル空間モデル」やエラーを許す文字列照合の手法である「編集距離」を使用する。計算方法を以下に示す。

【詳細プロトコル情報】「tf・idf法」を用いて、各パケットフローの詳細プロトコル情報に関するベクトル空間を作成する。「tf・idf 法」を使用することにより、あるパケットフローにおける出現頻度が高く、全てのパケットフローの中で特定のパケットフローに偏在する詳細プロトコル情報がそのパケットフローの特徴を表す情報であるとみなすベクトル空間を作成することが可能になる。そして、2 つのベクトル空間の距離を計算することで、類似度を算出する（文献[3]）。

【データ情報】2 つのパケットフローのデータ情報を同じにするために必要な挿入、削除、置換の合計数の最小値（編集距離）（文献[4]）を計算することで、データ情報の類似度を算出する。

【パケット並び順】2 つのパケットフローのパケット並び順を同じにするために必要な挿入、削除、置換の合計数の最小値（編集距離）（文献[4]）を計算することで、パケット並び順の類似度を算出する。

最後に、3 つの類似度の重み平均（式(1)）を計算することで、パケットフロー間の最終的な類似度を計算する。

最終的な類似度
＝（a×詳細プロトコル情報の類似度＋
ß×データ情報の類似度＋
γ×パケット並び順の類似度）/3　　式(1)

a：詳細プロトコル情報用パラメータ
ß：データ情報用パラメータ
γ：パケット並び順用パラメータ

4.4. 検知した未知ワームの自動分類

提案システムでは、統計分析結果を用いて、既知のワーム情報を登録することができる。登録された既知ワーム情報と作成されたパケットフロー間で類似度の計算を行うことで、既知ワームと似ているパケットフローの集合を作成することができる。これにより、未知のワームについて、それが既知のワームの亜種であるか否かを判断することが可能となる。

5. システムの実装及び動作検証

パケットフロー統計分析システムの実装を行い、提案方式の有効性の検証を行った。

5.1. 実装環境

実装環境を表 2に示す。

表 2 システム実装環境

OS	Fedora Core2
Web サーバ	Apache 2.0.40
アプリケーションサーバ	Tomcat 4.1.27
データベース	PostgreSQL 7.4.6
その他のアプリケーション	Tethereal 0.10.8

5.2. 検証用データ

動作検証に用いたデータは、2004 年 8 月～2005 年 1 月にインターネットに囮サーバ 2 台を接続して取得した。囮サーバの構築には、アプリケーションの挙動を模擬するために、honeyd0.8b を使用した。このときの囮サーバの設定を表 3に示す。

表 3 囮サーバの設定

アクセス条件	レスポンス内容
ICMP Echo	Echo Request
TCP80	Error ページの表示
TCP21	ログインプロンプトを表示し、anonymous ログインを許可
上記以外の TCP SYN	SYNACK
UDP	なし

5.3. 提案システムのワーム検知結果

検証用データを提案システムに投入し、検知内容を確認した。また、システムの類似度計算におけるパラメータを変化させたときの検知内容の変化を確認した。検知結果を表 4に示す。

表 4 検知したワーム及び不正アクセスの種類とその亜種の種類数

項番	名称	検知した亜種の種類数
1	MSBlaster とその亜種	6
2	Welchia	—
3	Sasser とその亜種	3
4	Gabot とその亜種	10
5	RAHack	—
6	DipNet/OddBob	—
7	CodeRed	—
8	Unicode 脆弱性攻撃	—
9	Windows のメッセジャサービスを利用したスパム広告	—
10	SQLSlammer	—
11	Portscan	—

これにより、提案システムでは、数年前に発生した「CodeRed」から、2005 年 1 月初旬に発生している「DipNet/OddBob」まで、様々なワームを検知することがでることが確認できた。また、パラメータを変化させることにより、ワームの亜種をまとめて表示することが可能になるので、ワームの亜種の発見しやすくなることが判明した。しかし、パラメータの設定によっては、検知できないワームがあることが判明した。

5.4. Snort のワーム検知結果

検証用データを Snort2.2.0（含まれているすべてのルールにおいて、該当する不正アクセスを検知した場合アラートが挙がるように設定）に投入し、アラート内容を確認した。Snort の検知結果と提案システムの検知結果を比較した結果を表 5 に示す。

これにより、提案システムを使用することで、Snort が検知しないワームについても検知が可能であることが判明した。

表 5 Snort 検知結果との比較

項番	名称	関連していると思われる Snort のアラート
1	MSBlaster とその亜種	なし
2	Welchia	ICMP PING
		WEB-MISC bad HTTP/1.1 request, Potentially worm attack
3	Sasser とその亜種	なし
4	Gabot とその亜種	BACKDOOR DoomJuice file upload attempt
		SHELLCODE x86 NOOP
5	RAHack	なし
6	DipNet/OddBob	なし
7	CodeRed	なし
8	Unicode 脆弱性攻撃	WEB-IIS scripts access
		(http_inspect) BARE BYTE UNICODE ENCODING
9	Windows のメッセジャサービスを利用したスパム広告	なし
10	SQLSlammer	なし
11	Portscan	SCAN SYN FIN
		SCAN synscan portscan
12	その他 (TCP 系)	(http_inspect) NON-RFC HTTP DELIMITER
		WEB-IIS nsiislog.dll access
		WEB-MISC Invalid HTTP Version String
		MISC source port 53 to <1024
		BAD-TRAFFIC tcp port 0 traffic
		BACKDOOR typot trojan traffic
13	その他 (UDP 系)	(snort_decoder): Short UDP packet, length field > payload length
		DNS named version attempt
		BAD-TRAFFIC bad frag bits

5.5. 動作検証に関する考察

動作検証をした結果、パラメータの設定によっては、発見できないワームがあることが判明した。そこで、効果的にワームを発見するためのパラメータ設定についての考察を以下に示す。

- 囮サーバの設定や設置環境、捕獲したいワームの種類によって、使用するパラメータを変化させる必要がある。そのため、IDS のシグネチャチューニングのように、使用前に、パラメータチューニング期間を設けるよいと考えられる。
- 今回使用したテストデータでは、複数のパラメータセットを使用すると、ワームが効果的に検知できることが判明した。その推奨パラメータセットを表 6 に示す。

表 6 推奨パラメータセット

項番	パラメータ		
	詳細プロトコル情報	データ情報	パケット並び順
1	1.0	1.3	1.3
2	0.0	1.0	0.0
3	0.0	0.0	1.0

6. まとめ

インターネット上の囮サーバで送受信されたパケットデータを統計分析することによって、未知のワーム及びワームの亜種を検知するシステムを提案した。そして、動作検証により、提案システムが囮サーバに対するワームアクセスを検知できることを証明した。今後は、バーチャルアプリケーションによる囮サーバではなく、本物のアプリケーションを使用した囮サーバによる提案手法の有効性を証明したい。また、リアルタイム検知機能等の機能追加を行い、提案システムがより効果的になる方策を検討したい。

文　献

[1] 独立行政法人　情報処理推進機構：未知ウィルス検出に関する調査、2004 年 4 月

[2] 山西健司, 竹内純一, 松永祐子：セキュリティマイニング, NEC 技法　Vol.56 No.12 2003

[3] David Dagon, Xinzhou Qin, Guofei Gu, Wenke Lee, Julian Grizzard, John Levine, and Henry Owen : HoneyStat : Local Worm Detection Using Honeypots, RAID 2004, LNCS3224, pp.39-58, 2004

[4] 大谷紀子：情報検索におけるベクトル空間モデルの応用, 武蔵工業大学環境情報学部紀要, 第五号, pp.99-109, 2004 年 2 月

[5] GONZALO NAVARRO : A Guided Tour Approximate String Matching, ACM Computing Surveys, Vol.33 No.1, March 2001