擬似 Private VLAN における Worm の感染防止と異常検知に関する研究

大阪産業大学大学院 工学研究科
情報システム工学専攻 情報安全工学研究室
03MH01 赤松康介

2004 年度

要旨

現在、Worm の感染を防ぐ手段として、いくつかの方法が提案されている。それらの中に、検疫ネットワークを用いた方法がある。検疫ネットワークを実現する方法は、いくつか存在する。しかし、その中の一部の方法では、ユーザが不正に設定を行なうことで、検疫を受けずにネットワークを利用することができる。また、それが不可能な方法の場合、機器の導入にコストが発生する。本研究では、全てのクライアントコンピュータに対して Worm の感染対策が有効であるようにし、なおかつコストを低く抑えられる方法を考えた。そのために DHCP(Dynamic Host Configuration Protocol) などを使って擬似 Private VLAN(Virtual Local Area Network) を構築した。これを用いることで、直接通信を行なうことによって感染する Worm が、組織内ネットワークの他のコンピュータに感染するのを防ぐことができる。

また、Worm に感染したコンピュータの存在を検知するために、次の 3 つの異常検知機能を付加した。まず、既存の IDS(Intrusion Detection System) を利用した異常検知機能である。これは既存の Worm が感染活動を行なう際の攻撃パターンに基き、異常を検知する。次に、おとり IP アドレスを利用した異常検知機能である。これは通常は通信が行なわれない IP アドレスに対する通信を監視することによって、異常を検知する。最後に、パケットが送信される先の数と、閾値を用いた異常検知機能である。これはパケットが送信される先の数を数え、それを基に異常を検知する。本研究で実現した異常検知機能は、多くのコンピュータに対して感染活動を行なうという Worm の性質を利用している。そのため、未知のセキュリティホールを利用して感染する未知の Worm であっても、検出が可能であると考えられる。

目 次

1 はじめに **1**

2 Worm について **2**

3 一般的な Worm 感染対策 **4**
3.1 Worm 感染対策 4
3.2 検疫ネットワーク 4
3.2.1 パーソナルファイアウォールを用いる方式 6
3.2.2 DHCP サーバを用いる方式 6
3.2.3 認証スイッチを用いる方式 7

4 本研究で提案する Worm 感染対策 **8**
4.1 擬似 Private VLAN 8
4.2 擬似 Private VLAN を使った例 9
4.3 異常検知 12
4.3.1 IDS による異常検知 12
4.3.2 おとり IP アドレスを用いた異常検知 13
4.3.3 パケットの送信先の数と、閾値を用いた異常検知 14

5 考察 **16**

6 まとめ **18**

A 作成したスクリプト **20**
A.1 “インタフェイス有効化スクリプト” 生成スクリプト 20
A.2 iptables 設定スクリプト 21
A.3 dhcpd 用設定ファイル生成スクリプト 22
A.4 パケット送信先の解析/閾値設定支援スクリプト 23

1　はじめに

現在、コンピュータは様々な危険にさらされている。その中の 1 つにコンピュータウィルスの感染がある。コンピュータウィルスが感染すると、外部への攻撃が行なわれたり、コンピュータの情報が破壊されたりするおそれがある。また、組織内の重要な情報が外部に漏洩してしまうおそれもある。コンピュータウィルスの中でも、ネットワークを使って感染するものを Worm という。コンピュータがネットワークに繋がっているだけで、Worm は爆発的に広がる。また、Worm の感染活動自体がネットワークに被害を与えることもある。2003 年には、“W32/MSBlaster” や “W32/Welchia” などの Worm が大流行した。これらは他のコンピュータに感染するために、多くの通信を行なう。このため、ネットワークが過負荷になり、通常の通信に支障をきたした。

Worm の感染経路としては、インタネットから直接攻撃を受けることは比較的少ない。組織内にある未対策のコンピュータや、外部から持ち込まれるコンピュータを経由して、感染が広まることの方が多い。このような Worm の感染に対して、次のような対策が現在行なわれている。まず、ウィルス対策ソフトウェア [1]のパターンアップデートおよび OS [2]のセキュリティアップデートを行なうようにセキュリティポリシ [3]を定める。常にこれらのアップデートを行なうことで、既知の Worm に感染する危険はなくなる。

しかしセキュリティポリシを定めても、実際に作業が行なわれるとは限らない。そこで最近では、これらのアップデートを徹底するために検疫ネットワークが用いられるようになってきた。検疫ネットワークを用いた対策では、コンピュータを組織内ネットワークに接続する前に、検疫を行なう。Worm に感染する可能性のある、脆弱なコンピュータに対しては、感染しないよう対策を行なう。これによって、組織内ネットワークへの感染を防ぐことができる。

検疫ネットワークを実現する仕組みは、いくつかある。中には、不正に検疫をすりぬけることができるものがある。また、それができないものの場合、機器の導入にコストが発生する。そこで本研究では、検疫ネットワークを用いなくとも Worm の感染を防止できる仕組みを提案する。

---

[1]コンピュータに感染したコンピュータウィルスを発見・駆除するためのソフトウェア。このソフトウェアを動かしておくことで、感染を阻止することもできる。既知のコンピュータウィルスの特徴をパターンファイルとして保存している。新たに見つかったコンピュータウィルスの情報を取得するために、定期的にパターンアップデートが必要である。

[2]Operating System. キーボードからの入力や画面への出力といった入出力機能やディスクやメモリの管理などの基本的な機能を提供し、コンピュータシステム全体を管理するソフトウェア。

[3]組織内のセキュリティに関する基本的な方針や行動指針のこと。

2 Worm について

Worm は、コンピュータウィルスの一種である。コンピュータウィルスとは、感染・潜伏・発病の機能を持ったプログラムのことをいう。コンピュータウィルスは実行されると、なんらかの形で他のコンピュータに感染しようとする。例えば、コンピュータに存在するファイルに自分自身 (コンピュータウィルス) をコピーする。ユーザがそのファイルを他のコンピュータにコピーし、実行することでコンピュータウィルスが感染する。またコンピュータウィルスは、発病の条件が揃うまでユーザに見つからないようにする。何もしないものもあるが、偽装を行なうものもある。ユーザが、感染したファイルのファイルサイズを表示しようとした際、感染前のファイルサイズを表示するといったことを行なうものもある。そしてコンピュータウィルスは、条件が揃うと発病する。発病すると、ユーザの意図しない、なんらかの処理が実行される。コンピュータ内の情報を削除するなどの破壊活動を行なうものや、特定のコンピュータに対して攻撃を行なったりするものがある。このようなコンピュータウィルスの中でも、ネットワークを使って他のコンピュータに感染するものを Worm という。

Worm は大きく分けると、以下の 2 種類に分けることができる (**図 1**)。

- 感染先のコンピュータに直接通信を行なうもの
- 間に別のコンピュータを介して通信を行なうもの

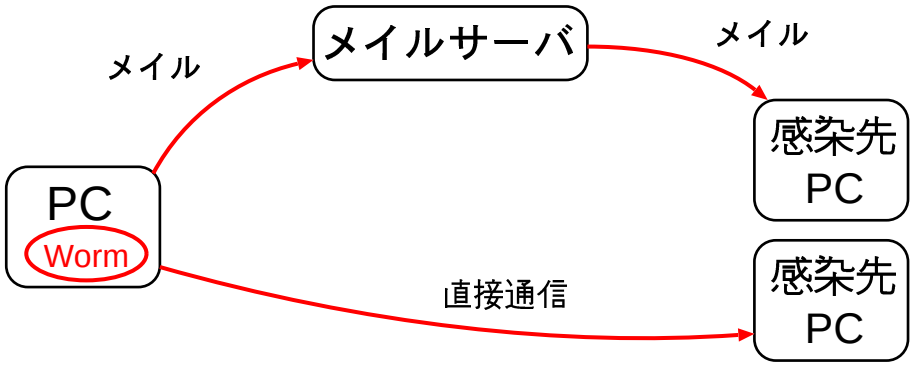

図 1: Worm の感染経路

前者は他のコンピュータに対して直接通信を行なう。利用するプロトコルは、HTTP [4] [1] [2] やファイル共有に使われるプロトコルなどである。そして OS や

[4]Hyper Text Transfer Protocol. World Wide Web サービスにおいて、サーバとクライアン

サーバソフトウェアのセキュリティホール [5] を利用して感染する。

後者は中継サーバを経由して通信を行なう。利用するプロトコルは、主にメイルの転送に使われるプロトコル (SMTP [6] [3]) である。メイルを介するものは、感染したコンピュータのアドレス帳などに保存されているメイルアドレスに対して、メイルを送信する。送信するメイルにファイルを添付し、受け取った人間がそのファイルを実行することで感染する。また、受け取ったメイルを表示すると、自動的に添付ファイルを実行するようなセキュリティホールを持つメイラが存在する。これを利用して感染するものもある。

トの間でデータを転送するために用いられるプロトコル。HTTP/1.0 は RFC1945 として、HTTP/1.1 は RFC2616 として定められている。

[5] ソフトウェアの設計ミスなどによって生じた、システムのセキュリティ上の弱点。

[6] 電子メイルを転送する際に用いられるプロトコル。RFC2821 として定められている。

3　一般的な Worm 感染対策

3.1　Worm 感染対策

現在、Worm の感染対策として、以下のようなことが行なわれている。

- ウィルス対策ソフトウェアの導入
  - 各クライアントへの導入
  - メイルゲイトウェイへの導入
- セキュリティアップデートの適用
- 検疫ネットワークを用いたアップデートの徹底

先に述べたように Worm には、メイルを介して感染するものと、他のコンピュータに対して直接通信を行なって感染するものがある。

メイルを介して感染するものについては、既存のメイルゲイトウェイなどによって防ぐことができる。これはメイルを中継する際にウィルススキャンを行ない、コンピュータウィルスや Worm を見つけると削除する。

他のコンピュータに対して直接通信を行ない、セキュリティホールを利用して感染するものについては以下の 2 つが必要となる。

- ウィルス対策ソフトウェアのパターンをアップデートし、最新の状態に保つ
- セキュリティアップデートを行ない、最新の状態に保つ

しかし、これらのアップデートを、外部から持ち込まれるコンピュータに対して徹底するのは困難である。これに対する対策として、検疫ネットワークを用いるものがある。検疫ネットワークを用いた対策は、先に述べたアップデートを徹底する方法である。

3.2　検疫ネットワーク

検疫ネットワークは、主に外部から持ち込まれるコンピュータに対する対策として利用される。おおまかに言うと、コンピュータの安全性が確認されるまでは組織内ネットワークに接続させない、というものである。検疫ネットワークを用いた対策では、コンピュータを組織内ネットワークに接続しようとすると、組織内ネットワークとは別にある、検疫専用のネットワークに接続される (**図 2**)。これが

検疫ネットワークと呼ばれる。そしてセキュリティアップデートが最新かどうか、ウィルス対策ソフトウェアのパターンが最新かどうかの 2 点が強制的にチェックされる。最新でなければ組織内ネットワークには接続されない。その場合、検疫ネットワークからアップデートファイルやパターンファイルを入手し、アップデートを行なう。チェックに合格すれば、組織内ネットワークに接続される (**図 3**)。

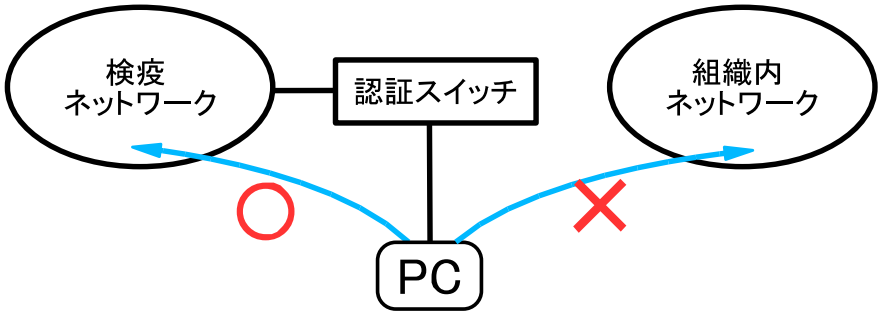

図 2: チェック合格前 (検疫ネットワークに接続されている状態)

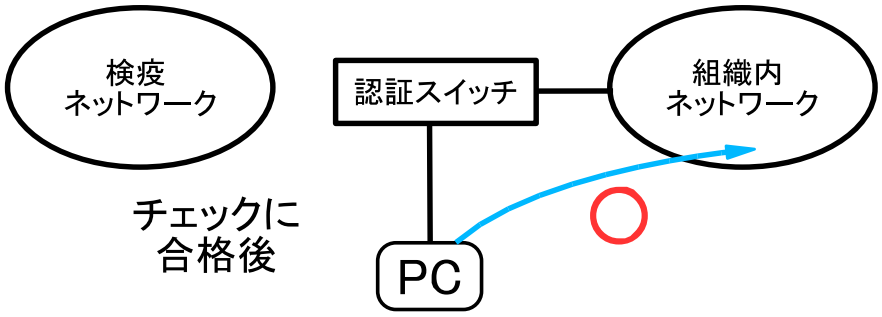

図 3: チェック合格後 (組織内ネットワークに接続されている状態)

これによって組織内ネットワークに繋がっている全てのコンピュータについて、セキュリティアップデートおよびウィルス対策ソフトウェアのパターンを最新に保つことができる。仮に、外部から持ち込まれたコンピュータの中にコンピュータウィルスや Worm が存在しても、既知の Worm は、全て検出することが可能になる。こうして組織内ネットワークへの感染を防ぐことができる。

検疫ネットワークを用いる場合は、チェック合格前と合格後で、接続先のネットワークの切り替えが必要となる。接続先のネットワークの切り替えには、以下の 3 つの方式がある。

- パーソナルファイアウォールを用いる方式

- DHCP [7] [4][5] サーバを用いる方式
- 認証スイッチ [8]を用いる方式

3.2.1 パーソナルファイアウォールを用いる方式

パーソナルファイアウォールを用いる方式は、クライアントコンピュータでパーソナルファイアウォール [9]を動作させる。チェック合格前は、検疫ネットワークにしか繋がらないように、ファイアウォールソフトウェアが設定される。チェックに合格すると、組織内ネットワークに繋がるように、ファイアウォールソフトウェアの設定が変更される。この方式の利点は、導入の際に組織内ネットワークの構成を変更する必要がないことである。しかし、この方式では、ファイアウォールソフトがインストールされていないコンピュータは、チェックを受けずに組織内ネットワークにアクセスできてしまう。

3.2.2 DHCP サーバを用いる方式

DHCP サーバを用いる方式は、IP アドレスで接続先を切り替える。チェック合格前は、検疫ネットワークにしかアクセスできない IP アドレスが配布される。チェックに合格すると、組織内ネットワークにアクセスすることのできる IP アドレスが配布される。この方式では従来のスイッチが使用可能であり、スイッチの交換は必要ない。しかし、この方式にも問題がある。組織内ネットワークにアクセスすることができる IP アドレスを、ユーザが DHCP を使用せずに手動で設定することができることである。これによって、チェックを受けずに組織内ネットワークにアクセスできてしまう。

[7]Dynamic Host Configuration Protocol. LAN 内のネットワーク機器に、IP アドレスなどの IP 設定情報を自動的に割り当てるためのプロトコル。

[8]ユーザを認証する機能を持つスイッチ。認証サーバなどと連携して、認証に合格したユーザだけがネットワークを利用可能にする機能を持つ。また、アクセス可能なスイッチのポートなどを、ユーザ毎に設定することができる。

[9]Worm による攻撃や、他のマシンからの攻撃を阻止するのに使われるソフトウェア。特定の IP アドレスからの通信を拒否するなどの、アクセス制御を行なうことができる。

3.2.3　認証スイッチを用いる方式

認証スイッチを用いる方式は、ユーザ認証の可能な認証スイッチを利用する。認証スイッチは、認証サーバなどと連携することによって、アクセス可能な VLAN [10] をユーザ毎に設定することができる。チェック合格前は、検疫ネットワークとなる VLAN にしかアクセスすることができない。チェックに合格すると、ユーザのアクセス可能な VLAN が変更され、組織内ネットワークの VLAN にアクセス可能になる。この方式は、チェックに合格せずに組織内ネットワークにアクセスする方法がなく、最も安全である。しかし、従来のスイッチを全て認証スイッチに置き換える必要があり、コストが発生する。

[10] Virtual LAN の略称。LAN において、物理的な接続形態とは独立に、コンピュータの仮想的なグループを設定すること。同一のスイッチに繋がったコンピュータ群を、複数のスイッチを用いて複数の独立した LAN を構成した場合と同様に運用することができる。

4　本研究で提案する Worm 感染対策

検疫ネットワークを用いる場合、不正に検疫をすりぬけることができるものがある。そして、それができないものの場合、機器の導入にコストが発生する。また、未知の Worm に対して無防備である。そこで本研究では考え方を変えてみた。検疫ネットワークの考え方は「Worm を組織内ネットワークに持ち込ませない」というものである。本研究では「Worm に感染したコンピュータがあっても、他のコンピュータに感染させない」という考え方に基き、擬似 Private VLAN を構築して Worm の感染を防止することにした。

第 3.1 節で述べたように、ウィルス対策ソフトウェアの導入や、セキュリティアップデートの適用は従来通り行なう。そして検疫は行なわない。検疫ネットワークを用いる代わりに、擬似 Private VLAN を用いて Worm の感染を防止する。

4.1　擬似 Private VLAN

Private VLAN [6] というものが考えられ、実用化されつつある。一部の高価なスイッチには、この機能がついている。Private VLAN とは、同一の VLAN 内においてクライアントコンピュータの間で直接通信をできなくする技術である。そしてクライアントコンピュータと特定のポートに繋がった機器 (ゲイトウェイ) との間の通信のみが許可される。これによって、クライアントコンピュータ間の直接通信を禁止しながら、インタネットやサーバなどへの通信を可能にしている。

しかし、そのようなスイッチの導入にはコストが発生する。そこでコストをかけずに同等の機能を実現した。ここではこれを、擬似 Private VLAN と呼ぶことにする。全てのクライアントコンピュータを論理的に別のネットワークに接続し、1 つのネットワークに 1 つのクライアントコンピュータとする。こうすることで、擬似 Private VLAN を実現した。

本研究ではゲイトウェイとなるコンピュータの OS として、GNU/Linux を用いた。GNU/Linux の iptables [11] [8] および IP エイリアスを利用した。そして DHCP サーバも、同一のコンピュータで動かした。具体的にはまず、 DHCP を使って、それぞれのクライアントコンピュータに、全て別のネットワークになるような IP アドレスを割り当てる。それぞれのネットワークにはゲイトウェイが必要となる。ここでは 1 台のコンピュータを、複数のネットワークのゲイトウェイとして機能させる。これには IP エイリアスを利用する。IP エイリアスを利用すれば、 1 つ

[11] GNU/Linux で使われている、パケットをフィルタするプログラム。パケットを、送信元や送信先などの情報を基に、設定したルールにそって判断し、破棄したり受け入れたりする。

のネットワークインタフェイスに複数の IP アドレスを割り当てることができる。そしてゲイトウェイとなるコンピュータで、パケット [12] の転送設定を行なう。ここでは iptables というソフトウェアを用いる。ファイルサーバなどやインタネット向けのパケットは転送し、他のクライアントコンピュータ向けのパケットは転送しないようにする。このようにして擬似 Private VLAN を構築する。

本研究の対策では、コンピュータを持ち込む際、チェックを行なわずに組織内ネットワークに接続する。そして擬似 Private VLAN を用いて、組織内ネットワークに接続しても、他のクライアントコンピュータに対して直接通信を行なうことができないようにする。ファイルサーバなどとインタネットへの接続のみを許可する (**図 4**)。これによって、セキュリティホールを抱えたコンピュータが存在しても、通信できないため Worm に感染することはない。

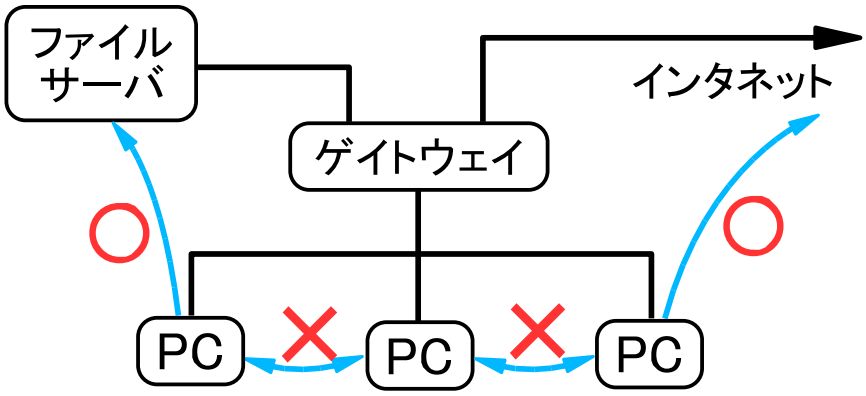

図 4: 擬似 Private VLAN

4.2 擬似 Private VLAN を使った例

ここでは、擬似 Private VLAN を構築した具体的な例について説明する。クライアントコンピュータは、192.168.2.0/24 のアドレスを利用することにする。そしてファイルサーバなどは、 192.168.1.0/24 のアドレスを利用することにする。ゲイトウェイとなるコンピュータには、ネットワークインタフェイスを 3 つ用意し、1 つをインタネットとの接続に使用する。もう 1 つをファイルサーバなどとの通信に使用し、192.168.1.0/24 のアドレスを割り当てる。そして残った 1 つをクライアントコンピュータとの通信用に用いる。IP エイリアスを利用して、このインタフェイスに複数の IP アドレスを割り当てる。

[12] データを転送する際に、データは小さなまとまりに分割されて転送される。その分割された小さなまとまりのこと。1 つ 1 つのパケットには宛先などの情報が付加されている。

ネットマスク [13] が 24 ビットのアドレスを使うとき、ネットマスク を 30 ビットにすれば64 の論理的なネットワークができ、64 台のクライアントコンピュータを接続することができることになる。このとき、1 つのネットワークあたりの IP アドレスは、4 つになる。ネットワークには、ネットワークアドレス [14]とブロードキャストアドレス [15]が必要である。これらは、特定のインタフェイスに割り当てて使用することはできない。そのため残りのアドレスは 2 つとなる。それらをクライアントコンピュータとゲイトウェイに割り当てることにする。

ここでは 192.168.2.0/24 のアドレスを利用した。すると、64 のネットワークができ、**表 1** のように IP アドレスを割り当てることができる。このように IP アドレスを配布するよう、DHCP サーバの設定を行なう。ゲイトウェイの IP アドレスは固定して割り当て、クライアントコンピュータの IP アドレスのみ DHCP を用いて配布する。ネットワークを構築する際、物理的には **図 5** のように接続する。しかし論理的には **図 6** のように動作する。 ゲイトウェイにおけるパケットの転送ルールは **表 2** のように設定する。クライアントコンピュータに割り当てられる IP アドレスは 192.168.2.0/24 のアドレスに含まれる。そこでクライアントコンピュータ同士の通信である、192.168.2.0/24 から 192.168.2.0/24 へ向かうパケットは転送しないようにする。その他の、ファイルサーバなどのある 192.168.1.0/24

表 1: IP アドレスの割り当て

ネットワーク	ネットワークアドレス	クライアントアドレス	ゲイトウェイアドレス	ブロードキャストアドレス
192.168.2.0/30	192.168.2.0	192.168.2.1	192.168.2.2	192.168.2.3
192.168.2.4/30	192.168.2.4	192.168.2.5	192.168.2.6	192.168.2.7
192.168.2.8/30	192.168.2.8	192.168.2.9	192.168.2.10	192.168.2.11
⋮	⋮	⋮	⋮	⋮
192.168.2.248/30	192.168.2.248	192.168.2.249	192.168.2.250	192.168.2.251
192.168.2.252/30	192.168.2.252	192.168.2.253	192.168.2.254	192.168.2.255

[13]IP アドレスからネットワークアドレスを求める場合に使用する値のこと。IP アドレスのうち、何ビットをネットワークを識別するためのネットワークアドレスに使用するかを定義する 32 ビットの数値である。ネットマスクには、通常は上位の側から連続した 1 を用い、たとえば 11111111111111111111111100000000 などとして、これを 24 ビットのネットマスクなどと呼ぶことがある。

[14]そのネットワーク自体を指し示す特殊なアドレス。

[15]ネットワーク内のすべてのコンピュータに対してデータを送信するために使われる、特殊なアドレス。

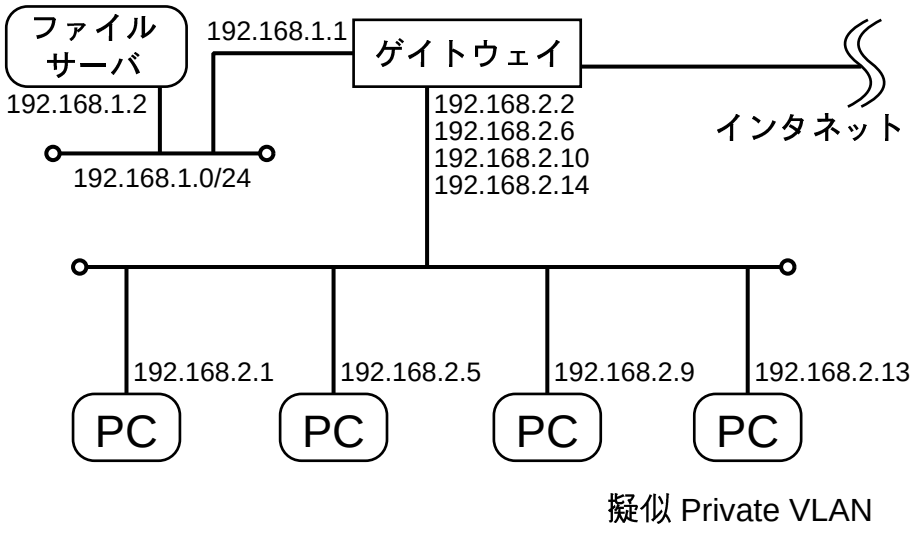

図 5: 物理的なネットワーク構成

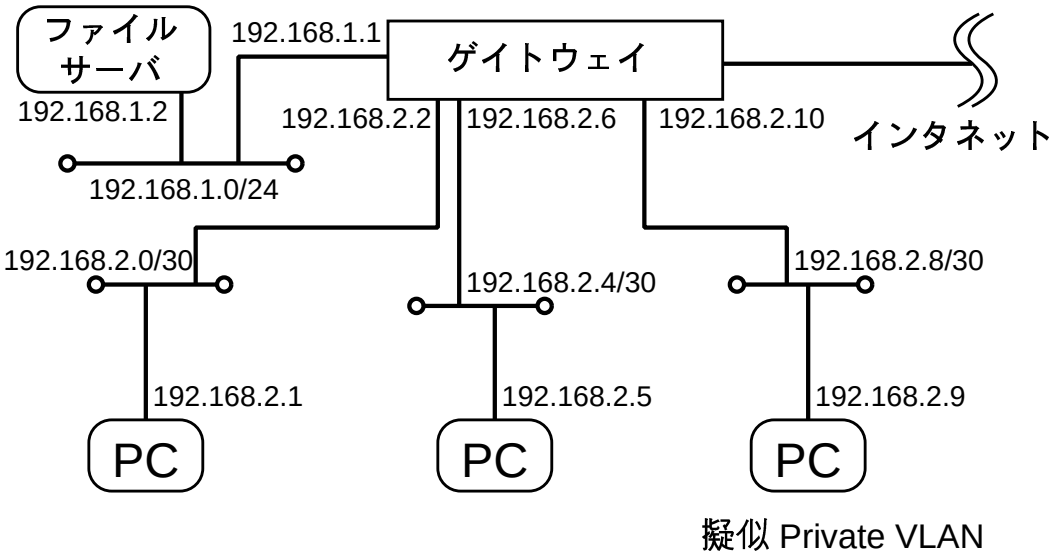

図 6: 論理的なネットワーク構成

表 2: パケット転送ルール

送信元	→	送信先	転送許可
192.168.2.0/24	→	192.168.2.0/24	不許可
192.168.2.0/24	→	192.168.1.0/24	許可
192.168.2.0/24	→	その他	許可

へ向かうパケットや、インタネット向けのパケットは転送するようにする。以上のような動作をするように iptables の設定を行なう。このような構成にすることで、クライアントコンピュータの間で直接通信を行なうことができなくなり、擬似 Private VLAN として動作させることができた。

4.3 異常検知

擬似 Private VLAN を用いて組織内ネットワークを構築した場合、Worm に感染しても気付きにくくなると考えられる。ウィルス対策ソフトウェアのパターンが古いままである可能性があり、さらに組織内ネットワークにある他のコンピュータにも感染しない。しかし外部への攻撃が可能であるため、なんらかの対策が必要である。そこで、ゲイトウェイに異常検知の機能を持たせることにした。

擬似 Private VLAN を使う場合、ネットワークの構成上、全ての通信はゲイトウェイを経由する。そこでゲイトウェイにおいて異常検知を行なう。本研究では、以下の 3 つの方法で異常検知を行なった。

- IDS [16] による異常検知
- おとり IP アドレスを用いた異常検知
- パケットの送信先の数と、閾値を用いた異常検知

4.3.1 IDS による異常検知

Worm は、セキュリティホールを利用して感染する。そこで Worm による感染活動を検出するために、異常パターン検知型の IDS を利用する。異常パターン検知型の IDS は、既知のセキュリティホールに対する攻撃パターンをデータベースとして持っており、パターンに合致する通信を検出する。Worm 自体ではなく、Worm による攻撃パターンを検出する。そのため、既知のセキュリティホールを利用して感染する Worm であれば、未知の Worm であっても検出が可能である。

本研究では、異常パターン検知型の IDS として Snort [7] を用いた。擬似 Private VLAN では、ネットワークの構成上、全ての通信はゲイトウェイを通過する。そこで、ゲイトウェイに IDS を設置し、ゲイトウェイを通過するパケットを監視する。

[16] Intrusion Detection System. 監視対象機器への侵入を検知するシステム。侵入の兆候や攻撃を検知すると、警告を発する。

4.3.2　おとり IP アドレスを用いた異常検知

通常、未使用の IP アドレスに対して通信を行なうことはない。しかし Worm は感染先のコンピュータを探すために、未使用の IP アドレスに対しても通信を行なうことがある (**図 7**)。このような性質を利用して Worm を検知できると考えられる。そこで、未使用の IP アドレスをおとりとして用意しておき、その IP アドレスに対する通信を監視する。通信があれば、通信を行なったコンピュータが異常であると判断することができる。

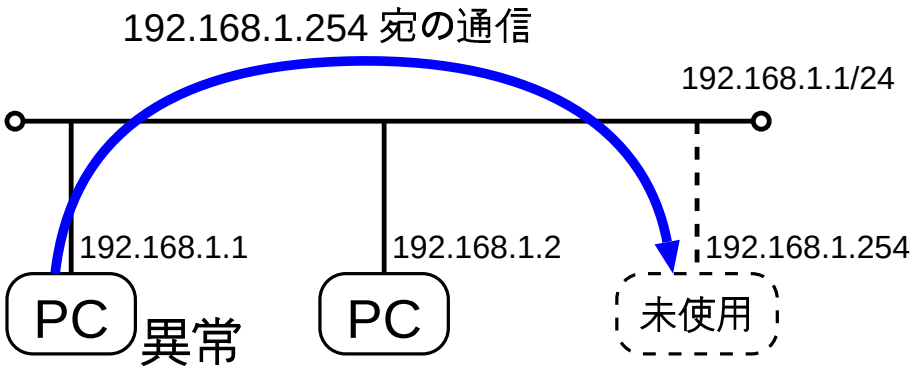

図 7: 異常なコンピュータの通信

監視対象となる、おとり IP アドレスを全体で 1 つ用意するか、各ネットワークに用意するのか考える。それには Worm が、自分の所属するネットワークの IP アドレスのみについて感染先のコンピュータを探すのか、ネットワークを越えて自分の所属しないネットワークの IP アドレスに対しても通信を行ない、感染先のコンピュータを探すのかが問題となる。前者である場合、各ネットワークに、監視対象となる、おとり IP アドレスを用意する必要がある。一方、後者である場合、全体で 1 つ用意すれば良い。Worm の目的を考えると、より多くのコンピュータに感染するために後者である可能性が高い。しかし、そうでない可能性もある。各ネットワークに、おとり IP アドレスを用意しておけば、どちらにも対応可能であるが、2 つ問題がある。

まず、多数の IP アドレスが無駄になるという点である。第 4.2 節では、ネットマスク を 30 ビットにした。ネットマスク を 30 ビットにすると、1 つのネットワークあたりの IP アドレスは 4 つになる。これは、**表 1** のように全て使用している。おとり IP アドレスを用意するためには、ネットマスクを 29 ビットにする必要がある。こうすることで、1 つのネットワークあたりの IP アドレスは 8 つになる。**表 3** のように 4 つの未使用の IP アドレスが用意できる。しかし、接続可能な

コンピュータの台数が半減してしまう。ネットマスク が 24 ビットのアドレスを使うとき、ネットマスク を 30 ビットにすれば、64 台のクライアントコンピュータを接続することができる。しかし、ネットマスク を 29 ビットにすると、32 台のクライアントコンピュータしか接続することができなくなる。

表 3: それぞれのネットワーク内における IP アドレスの割り当て

アドレス	29 ビット	30 ビット
0	ネットワーク アドレス	ネットワーク アドレス
1	クライアント アドレス	クライアント アドレス
2	ゲイトウェイ アドレス	ゲイトウェイ アドレス
3	未使用	ブロードキャスト アドレス
4	未使用	ネットワーク アドレス
5	未使用	クライアント アドレス
6	未使用	ゲイトウェイ アドレス
7	ブロードキャスト アドレス	ブロードキャスト アドレス

また、未使用の IP アドレスがあると、ユーザが DHCP を使わずに固定的に IP アドレスを設定することができてしまう。すると、1 つのネットワークに複数のクライアントコンピュータが存在することになり、Worm の感染が可能となる。これについては、ゲイトウェイのインタフェイスに IP アドレスを割り当てておくことで、回避できる。しかしゲイトウェイに割り当てられる IP アドレスが、非常に多くなってしまう。

本研究ではこれらの問題を考え、IP アドレスを効率良く利用できるように、おとり IP アドレスを全体で 1 つ用意することにした。ファイルサーバなどを設置しているネットワークに属する、未使用 IP アドレスの 1 つを、おとり IP アドレスとする。

この方法は、多くのコンピュータに対して感染活動を行なうという、Worm の性質を利用している。そのため、未知のセキュリティホールを利用して感染する未知の Worm であっても、検出が可能であると考えられる。

4.3.3 パケットの送信先の数と、閾値を用いた異常検知

Worm は、より多くのコンピュータに感染しようとする。そのために、より多くのコンピュータに対してアクセスし、感染先のコンピュータを探す。このアクセスは、通常の利用による通信よりも、はるかに多いと考えられる。この特徴を

利用して、Worm の感染活動を検知することにした。

当初は単純に、送信されるパケットの数を集計して異常を検知しようと試みた。通常時にあらかじめ、1 日に送信されるパケットの数からコンピュータ 1 台あたりの閾値を、各プロトコルについて決めておく。そして 1 日毎に送信パケットの数を集計し、閾値を超えるパケットを送信したコンピュータは異常であると判断する。しかし、動画のストリーミング再生やファイルの転送など、多くのデータをやりとりする場合、大量のパケットが送信される。これらが大きなノイズとなった。Worm が大量にパケットを送信しているのか、正常な通信なのかが区別できなかったのである。

そこで、送信パケットの数を集計するのではなく、パケットの送信先の数を集計することにした。IP アドレスとポート番号[17]をセットにして送信先とし、いくつの送信先に対してパケットを送信したかを集計する。同じ IP アドレス、ポート番号に対するパケットが、いくつ送信されていても 1 つと数える。こうすることで、特定のサービスで大量のデータを送信しても 1 つの通信とみなすことができる。そして、あらかじめ通常時におけるパケットの送信先の数から閾値を決めておく。閾値を超える送信先に対してパケットを送信したコンピュータがあれば、そのコンピュータは異常であると判断する。**表 4** では、それぞれたくさんパケットが送信されているが、送信先は 10.1.1.1:22, 10.1.1.1:80, 10.1.1.2:80 の 3 種類である。

表 4: 送信されたパケットの例

送信先 IP アドレス	送信先ポート番号	送信パケットの数
10.1.1.1	22	100
10.1.1.1	80	1000
10.1.1.2	80	300

この方法は、多くのコンピュータに対して感染活動を行なうという Worm の性質を利用している。そのため、未知のセキュリティホールを利用して感染する未知の Worm であっても、検出が可能であると考えられる。

[17]インタネット上の通信において、複数の相手と同時に通信を行なうために設けられた、補助的な値のこと。ポート番号を利用することで、1 つの IP アドレスで複数のサービスを提供することが可能になる。一般に利用するサービス (プロトコル) 毎にポート番号が決められている。例えば電子メールのサービス (プロトコル : SMTP) を利用する際には 25 の値が使われる。

5　考察

本研究での対策によって、組織内のクライアントコンピュータにネットワークを経由して Worm が感染する可能性は、ほぼ無くなった。しかし、**表 5** のように、Worm が組織外のコンピュータに対して感染活動を行なうことは可能である。この感染活動を異常検知機能によって検知する。検知可能な Worm は、**表 6** のとおりである。

表 5: Worm への対処方法

Worm の感染方法	感染先 (攻撃先)		対処方法
メイル経由	組織内の他のクライアント	→	メイルゲイトウェイで削除可能
	組織外のコンピュータ	→	メイルゲイトウェイで削除可能
直接通信	組織内の他のクライアント	→	擬似 Private VLAN で対処可能
	組織外のコンピュータ	→	対処不可能

表 6: 検知可能な Worm

検知対象		IDS	おとり IP	送信先の閾値
既知の Worm	既知のセキュリティホールを利用	○	○	○
未知の Worm	既知のセキュリティホールを利用	○	○	○
	未知のセキュリティホールを利用	×	○	○

本研究の対策を用いる利点としてまず、クライアントコンピュータを利用するユーザの負担が無いことが挙げられる。ユーザは、従来のネットワークと同じ手順で利用することができる。検疫ネットワークを用いる場合は、各クライアントにソフトウェアをインストールしたり、ネットワークに接続する際に認証が必要になったりする。これにより、ユーザの手間が増える。

また、導入コストが低いことが挙げられる。本研究の対策の場合、従来のスイッチをそのまま利用することができる。これによって、既に運用されているネットワークへの導入を、比較的、容易に行なうことができると考えられる。検疫ネットワークを用いる場合は、従来のスイッチを全て認証スイッチに置き替える必要がある。

本研究の対策を用いる欠点としては、Worm に感染しても気付きにくい点が挙げられる。擬似 Private VLAN を使用した場合、ウィルス対策ソフトウェアのパ

ターンが古いままである可能性があり、さらに周りのコンピュータにも感染しない。しかし組織外のネットワークへの攻撃は可能である。これに対する対策として、異常検知機能を付加することとした。

また、ゲイトウェイやファイルサーバなどに対して、クライアントコンピュータからアクセスが可能である。つまり、Worm の感染が可能である。そのため、ゲイトウェイやファイルサーバを管理する際には、ウィルス対策ソフトウェアのパターンを常に最新にし、OS やサーバソフトウェアのセキュリティアップデートを常に行なうよう、注意する必要がある。

擬似 Private VLAN では、全ての通信がゲイトウェイを経由する。そのため、ゲイトウェイとなるコンピュータの性能によって、従来のネットワークよりも転送速度が低下するおそれがある。

6　まとめ

擬似 Private VLAN と異常検知機能を組み合わせて、Worm の感染を防止するネットワークが構築できた。擬似 Private VLAN によって、Worm が組織内ネットワークに持ち込まれても、他のクライアントコンピュータに被害が広がらなくなる。また異常検知機能を利用することで、持ち込まれた Worm を発見することができる。

擬似 Private VLAN では、全ての通信がゲイトウェイを経由する。そこで、異常を検知した際に、異常のあるコンピュータからの通信を遮断する、自動遮断機能についても検討していく。現在の閾値を用いた異常検知機能は、通常時のパケットの送信先が増えると、それに合わせて調整する必要がある。今後は閾値の調整を容易にする方法についても検討する必要がある。また、異常なパケットの流れを判別するための、より良いアルゴリズムについても検討していく必要がある。

謝辞

本研究を進めていく上で、藤井 信夫教授、大垣 斉講師、中村 孝講師には御指導及び御協力を戴きました。また、高松 忍教授、樋口 清伯教授には副査として種々御助言を賜りました。fken.a4w メイルリストのメンバの方々には御助言を賜りました。ここに深く感謝の意を表します。

参考文献

[1] RFC1945, Hypertext Transfer Protocol – HTTP/1.0

[2] RFC2616, Hypertext Transfer Protocol – HTTP/1.1

[3] RFC2821, Simple Mail Transfer Protocol

[4] RFC2131, Dynamic Host Configuration Protocol

[5] RFC2132, DHCP Options and BOOTP Vendor Extensions

[6] Private VLANs: Addressing VLAN scalability and security issues in a multi-client environment (draft-sanjib-private-vlan-02.txt)

[7] Snort.org, [http://www.snort.org/]

[8] The netfilter/iptables project, [http://www.netfilter.org/]

※ URL はすべて 2005 年 1 月 7 日時点のもの

A　作成したスクリプト

A.1　“インタフェイス有効化スクリプト” 生成スクリプト

— ppvlan_ifup_gen.sh begin —

```
#!/bin/sh
for i in ‘seq 0 \‘expr 256 / 4 - 1\‘‘
do
  gw=‘expr $i \* 4 + 2‘
  bcast=‘expr $i \* 4 + 3‘
  if [ $i -eq 0 ]
  then
    iface=’eth1’
  else
    iface="eth1:‘expr $i - 1‘"
  fi
  echo -n "ifconfig ${iface} 192.168.2.${gw} "
  echo    "netmask 255.255.255.252 broadcast 192.168.2.${bcast} up"
done
```

— ppvlan_ifup_gen.sh end —

A.2 iptables 設定スクリプト

— ppvlan_iptables.sh begin —

```
#!/bin/sh


echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
iptables -P FORWARD DROP
iptables -A FORWARD -i eth1 -j LOG --log-prefix "IPTABLES-LOG: "
iptables -A FORWARD -s 192.168.2.0/24 -o eth0 -j ACCEPT
iptables -A FORWARD -d 192.168.2.0/24 -i eth0 -j ACCEPT
```

— ppvlan_iptables.sh end —

A.3 dhcpd 用設定ファイル生成スクリプト

— ppvlan_dhcpd.conf_gen.sh begin —

```
#!/bin/sh
echo 'shared-network ppvlan-dhcp {'
echo '  option domain-name "fken";'
echo '  option domain-name-servers 192.168.1.2;'
echo ''
for i in `seq 0 \`expr 256 / 4 - 1\``
do
  net=`expr $i \* 4`
  client=`expr $i \* 4 + 1`
  gw=`expr $i \* 4 + 2`
  bcast=`expr $i \* 4 + 3`
  echo "  subnet 192.168.2.${net} netmask 255.255.255.252 {"
  echo "    range 192.168.2.${client} 192.168.2.${client};"
  echo "    option routers 192.168.2.${gw};"
  echo "    option broadcast-address 192.168.2.${bcast};"
  echo '  }'
done
echo '}'
```

— ppvlan_dhcpd.conf_gen.sh —

A.4 パケット送信先の解析/閾値設定支援スクリプト

— count.rb begin —

```
#!/usr/bin/env ruby

# iptables -A FORWARD -i eth1 -j LOG --log-prefix "IPTABLES-LOG: "
# で出力した log を加工する filter

logid='IPTABLES-LOG'

counter = Hash.new()
while gets
   next if ! /#{logid}: /

   h = Hash.new()
   $_.sub(/^.* #{logid}: /,'').split(' ').each { |e|
      tag, value = e.split('=')
      value = 't' if value == nil
      h[tag] = value
   }

   str=''
   str << "PROTO=#{h['PROTO']}"
   if h['PROTO'] == 'ICMP' then
      str << " TYPE=#{h['TYPE']}"
   else
      str << " DPT=#{h['DPT']}"
   end
   str << " DST=#{h['DST']}"
   str << " SRC=#{h['SRC']}"
   if counter[str] == nil then
      counter[str] = 1
   else
      counter[str] += 1
```

```
   end
end
#p counter

# SRC 毎にまとめる。
c_num = Hash.new()
counter.each { |item|
   h = Hash.new()
   item[0].split(' ').each { |e|
      tag, value = e.split('=')
      value = 't' if value == nil
      h[tag] = value
   }
   str=''
   str << "#{h['DST']}"
   str << "#{h['PROTO']}"
   if h['PROTO'] == 'ICMP' then
      str << "#{h['TYPE']}"
   else
      str << "#{h['DPT']}"
   end
   str << " OUT=#{h['OUT']}"
   c_num[h['SRC']] = Hash.new() if !(c_num[h['SRC']])
   c_num[h['SRC']][str] = item[1]
}

# 表示
dests=[]
c_num.each { |host|
   dests << host[1].size
   print host[0],":",host[1].size,"\n"
}
print "max:",dests.max,"\n"
```

— count.rb end —