ボットの二次被害を防止する方法の提案

阿　南　宏　教

近年，ボットウィルスによる SPAM メールの送信や DDoS 攻撃，情報の奪取など様々な問題が蔓延している．ボットはオープンソースとなっているため亜種が多く存在し，検出するのが難しいという特徴がある．本研究ではプロセスツリーと送信時のマウス操作を監視することによりメーラを呼び出したのが正規なユーザか否かを判別する．正規なユーザと確認できた場合にのみ，ポートを開放しメールの通信を許可する．

Hiromichi Anan

Suggestions on how to prevent secondary damage of the bots

In recent years, sending email and SPAM Bottouirusu DDoS attacks, widespread abduction issue and various other information. There are many variants of bots is open-source because it has the characteristic that it is difficult to detect. In this study, we controlled the port, to determine whether the user is legitimate by calling the mailer to monitor and mouse operation transmission process tree. If so only users and regular email communications to allow for open ports.

1.　はじめに

インターネットの発展に伴い，ウィルスの被害が大きな問題となっている．近年では，ボットネットによる SPAM メールの送信や DDoS 攻撃，情報の奪取など様々な問題が蔓延している．ボットはオープンソースとなっているため亜種が多く存在する．現在蔓延するボットは，アングラサイトなどでソースコードが公開され，誰でも比較的簡単にボットの亜種を作り出すことが可能になった．一部ではボットの開発キットまで公開され，亜種の作成がさらに簡単になっている．亜種が生み出されるということは，対策する側にとって，それらの亜種に対してもパッチを配布するなど細かな対応をしなければならない．毎日のように提供される Windows アップデートも，これらのマルウエアに対応するためのものが少なくない．また，ボットは攻撃者の指示があるまで待機しているため，ボットに感染した場合，感染前との差異を感じることなくコンピュータを使用できるので，感染したことに気づきにくいといった問題もある．

本稿では，ボットによる感染は完全に防止できないという前提にたち，ボットが Herder の命令を受けて初めて行動を起こすことに着目し，クライアント側のスパムメール対策を施す．クライアントからメール送信される時に，プロセスツリーとマウスの操作を監視することにより正常なメール送信か否かを判断し，ポート制御を行うことによりボットによるスパムメール送信を遮断する方式を提案する．

以降，2 章で既存技術とその課題．3 章で提案方式とその動作について説明する．4 章で今後の検討課題を述べ，5 章でまとめる．

2.　既存技術とその課題

2.1 パターンマッチング．

ウィルスの種類ごとにウィルスがもつ特徴をデータとして集めておき，1 つずつ特徴を照らし合わせていくものである．しかし最近よく流行するウィルスは，自身の圧縮や暗号化などによって，パターンマッチングの検出にひっかからないようにするものもある．これに対し，ウィルス対策ソフト側も複雑なパターンマッチングを使って検査漏れを防いでいる．既に名前が知られているウィルスに対しては高い精度で駆除でき

るが，ウィルス情報がまったくないウィルスは検知することができない．

2.2 ヒューリスティック検知

実行ファイルなどに感染したウィルスを発見するには，ウィルス定義ファイルに列挙された既知のウィルスのパターンと照合していくのが一般的である．しかしこの方法では未知のウィルスや既存のウィルスを部分的に改変した亜種は発見できないといった問題やウィルスではないものをウィルスと誤認する問題がある．そこで，ウィルスを探索する際に実行ファイルの挙動などを解析し，ウィルスに特徴的な動作の有無を調べる手法が検出されている．これにより，未知のウィルスや亜種などにも対応できる可能性がある．

2.3 ハニーポット

ハニーポットの仕組みを図１に示す．セキュリティ的に問題のあるサーバやネットワークをインターネットにさらしておく．それらを監視し，調査することによって，攻撃者の手法や侵入者の行動を研究することができる．重要なサーバとは別にハニーポットを設置しておくと，攻撃者は比較的脆弱なハニーポットへ向けられる．重要なサーバへの攻撃を回避し，攻撃者の行動を把握するという目的もある．

ハニーポットによって，侵入者，攻撃者の行動はすべて筒抜けになる．そのため，ハニーポットがネットワークに設置されているかもしれないという心理が侵入者，攻撃者に働いた場合は，不正アクセスを抑止する効果もあるといわれている．しかし，ハニーポットの管理が甘かったりすると，そのサーバ自体が不正アクセスの踏み台に使われる危険性がある．そのため，安易にハニーポットを設置するのも危険である．

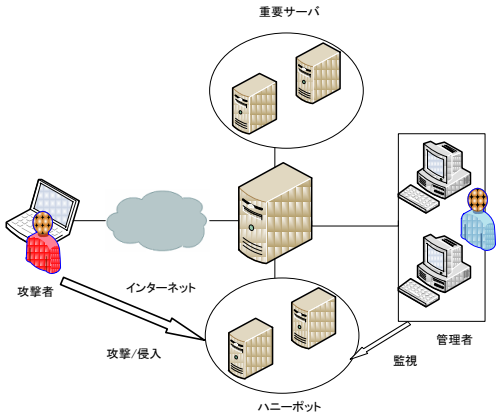


図 1 ハニーポットの仕組み

3. 提案方式

3.1 提案方式の流れ.

提案方式のメール送信までの流れを図２に示す．

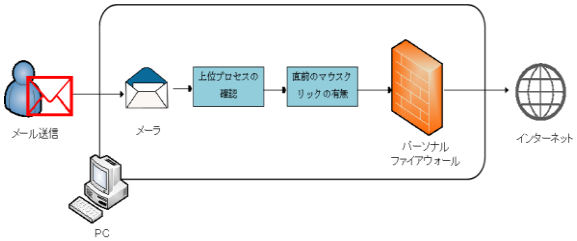


図 2 提案方式のメール送信までの流れ

ユーザがメール送信の命令を出した際，メーラの確認をプロセスツリーを用いて確認する．次に直前のマウスクリックがあったかどうか確認し，メールの送信命令を出したのが正規ユーザと判断できた場合のみポートを解放しメールを送信可能にする．3.2 より部分的に抽出して説明していく．

3.2 MAPI(Messaging API).

一般的に Windows 上では MAPI によりメールを送信する．MAPI は Windows 上で電子メールを扱うための標準仕様でメールメッセージを作成，転送するための関数郡である．

3.3 パーソナルファイアウォールの利用.

アンチウィルスソフトの機能としてパーソナルファイアウォールが含まれているものがある．パーソナルファイアウォールは，外部のネットワークからの侵入およびコンピュータ内部から外部ネットワークへの異常な通信を検知または遮断する．

ユーザが定義したルールに従って，パケットやプロトコルを許可または拒否することができる．(図 3) また，細かいポート制御も可能である．一般にメールを送信する際，SMTP ポート 25，587 番を使用する．提案方式では，パーソナルファイアウォールの設定で常に SMTP ポートを遮断しておく．メーラを呼び出したのが正規なユーザと確認できた場合にのみポートを開放する．通信終了後に再度ポートを遮断する．この方法により不正なメール送信を防止する．

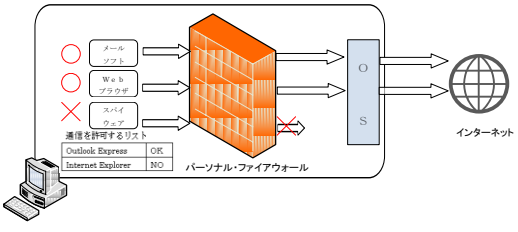
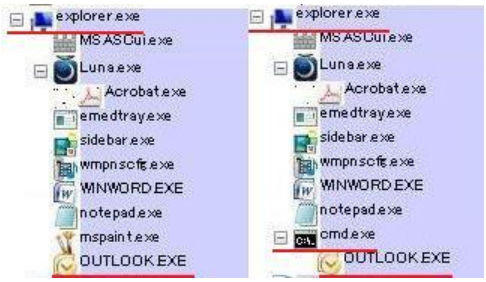


図 3 パーソナルファイアウォールによる遮断.

3.4 プロセスツリーによる上位プロセスの確認.



(1)正常時　　(2)異常時

図 4 プロセスツリーによる上位プロセスの確認

3.2.で述べたメーラを呼び出したのが正常なユーザかどうかを判断するためにプロセスツリーを監視する．プロセスツリーとは，実行中のプロセスをツリー上に表現したものである．メーラが実行されたとき，通常は explorer.exe が上位プロセスとなる．図 4 はプロセスを可視化するアプリケーションである Process Explorer v11.04 を用いて，プロセスツリーを表示したものである．メーラのプロセス名は outlook.exe である．正常時にはメーラの上位プロセスは explorer.exe となる．

一方，ボットに感染していると，メーラを呼び出す上位プロセスは正常時とは異なる．

図 4.(2) では，メーラを呼び出しているのが cmd.exe であり，異常とみなせる．図 4 は一つの例であり，ボットに感染していれば必ずメーラの上位プロセスが cmd.exe になると限らない．プロセスツリーにより，メーラの上位プロセスが explorer.exe と確認できた場合は正規なユーザがメーラを呼び出したと判断し，explorer.exe 以外の場合は不正なプログラムがメーラを呼び出したと判断できる．

3.5.プロセスモニター（Process Monitor）.

Process Monitor は，ファイルシステム，レジストリ，プロセスおよびスレッドの活動をリアルタイムで表示する，Windows 向けのツールである．ボットから SPAM メールが送信される時，遠隔操作により攻撃命令が送られる．そこでユーザが送信要求の有無を調べるため送信時のマウスクリックがあったかどうかを調べる．

プロセスモニターにより，ユーザの送信要求と確認できた場合は正規なユーザがメーラを呼び出したと判断し，異なる場合は不正なプログラムからのメールの送信要求があったと判断できる．

3.5 提案方式の動作.

提案方式の動作について図 5 に示す．本提案の動作

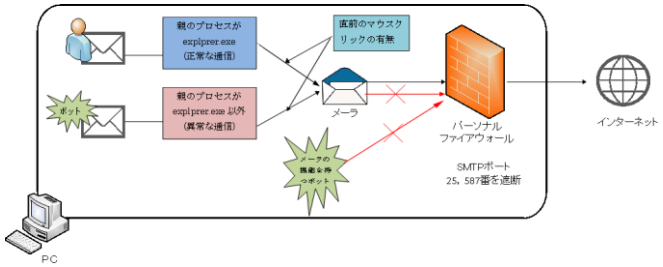


図 5 提案方式の動作

では，通常時はパーソナルファイアウォールのポート制御機能を利用し，SMTP ポート 25，587 番を遮断しておく．このため MAPI の関数群をフックし PC の動作を監視する．送信命令があった場合はプロセスツリーと送信直前のマウス操作を監視し，メーラを呼び出したのが正常なユーザであると確認できた場合にのみ，ポートを開放し通信終了後にポートを再度遮断する．

提案方式を実現するために，MAPI をフックし，プロセスツリーとプロセスモニターの検査を行う監視プログラムが必要である．監視プログラムはコンピュータが起動したときに同時に起動する．

MAPI 関数は 19 種類あり，その中でメール送信に使用される関数を表 1 にしめす．監視プログラムでは MAPILogon と MAPISendMail を監視する．

表 1　メール送信に使用される MAPI 関数

セッション名	機能
MAPILogon	メールサーバーへログオン．ユーザ名とパスワードを指定し，成功時にセッションハンドルを返す．
MAPILogoff	メールサーバからログオフ. MAPILogon にて返ってきたセッションハンドルを指定する．
MAPISendMail	MAPIMessage 構造体のメールコンテンツを送信する．

MAPILogon が呼び出された時にメーラが起動したと判断できるため，まず呼び出されたメーラが登録してあるメーラと一致するかを確認する．一致しなかった場合は不正なプログラムが動作している恐れがあり，ユーザにアラームをあげて，ポートを開放しない．呼び出されたメーラと登録してあるメーラが一致した場合は，メーラの上位プロセスをプロセスツリーにより確認する．メーラの上位プロセスが explorer.exe 以外のプログラムだった場合，不正なプログラムが動作

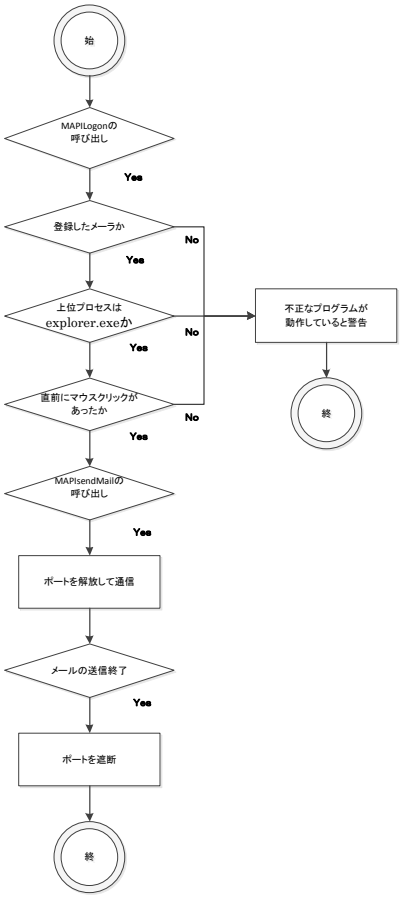


図 6 監視プログラムのフローチャート 5

している恐れがあるため, ユーザにアラームをあげて, ポートを開放しない．メーラの上位プロセスが explorer.exe の場合は，次にメール送信時の直前に送信ボタンのマウスクリックがあったかどうかをプロセスモニター-確認する．マウスクリックがなかった場合は, ユーザにアラームをあげて，ポートを開放しない．マウスクリックがあった場合は，次に MAPISendMail が呼び出されるのを待つ．メールの通信要求がされる時に MAPISendMail が呼び出されるので，この時点でポートを開放する．メール送信の終了を確認したら再びポートを遮断する(図 6).

以上によりボットによる SPAM メール送信を防止しユーザに対して危険にさらされていることを知らせることができる.

4. 今後の検討課題

本稿では端末からの不正なメール送信を防止し，ユーザに対して危険にさらされていることを警告する対策を提案した．しかし MAPI に対応していないメーラを使ったメール送信はポートが開放されず通信を行うことができないという課題がある．本研究では，使用するメーラが MAPI に対応している必要があるため今後検討が必要である.

5. まとめ

ボットにより PC がスパムメールを送信することを防止する対策として，プロセスツリーと直前のマウスクリックを監視し，メーラを呼び出したのが正しいユーザと判断できた場合にのみ，パーソナルファイアウォールの SMTP ポートを開放する手法を提案した. 今後は有効性を確認するために実装の検討を行う.

6. 参考文献

[1]“インターネット上の脅威「ボット」”
(http://itpro.nikkeibp.co.jp/free/ITPro/OPINION/20041216/153951)
[2]“ハニーポットとは”
(http://www.atmarkit.co.jp/fsecurity/special/13honey/honey01.html)
[3]“不正メールの送信防止とボット感染検知の検討”
(http://www.wata-lab.meijo-u.ac.jp/file/gthesis/2007/2007-GT_Abst-Ryoichi_Maiya.pdf)
[4]“MAPI とは”
(http://www.weblio.jp/content/MAPI)
[5]“パーソナルファイアウォールについて”
(http://support.microsoft.com/kb/321050/ja)
[6]“Process Explorer v11.04”
(http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx)
[7]“Process Monitor”
(http://technet.microsoft.com/ja-jp/sysinternals/bb896645)
[8]“API_Hook.zip”
(http://ruffnex.oc.to/kenji/text/api_hook)

7. 謝辞

本研究を進めるにあたり，多大なるご指導，ご鞭撻を賜りました渡邊晃教授に心より感謝いたします．また有益なご助言，ご検討を頂きました渡邊研究室の皆さんには深く感謝いたします.