IPS を用いた IRC ボットの活動抑制システムの検討

酒井 亮佑 †　　小林 孝史 ‡

†‡ 関西大学総合情報学部
569-1095 大阪府高槻市霊仙寺町 2-1-1
{k309342,taka-k}@kansai-u.ac.jp

あらまし 近年，マルウェアによる被害は増加しており，またマルウェア製作者の意図も愉快犯的なものから攻撃を行うことによって得られる利益を目的としたものに変わってきている．このような状況において，マルウェアによる被害はより深刻なものになると考えられる．そこで，本研究ではマルウェアの一種であるボットが，中継サーバ (C&C サーバ) と通信を行うときによく利用される IRC プロトコルにおいて，正規の IRC クライアントが行う通信に識別子をつけることによって，ボットが行う通信と区別する．また，ボットによる通信を侵入防止システム（IPS）を用いて遮断することでボットの活動を抑制できることを目指す．

An apploach for controling IRCbot's activity using IPS

Ryosuke Sakai†　　Takashi Kobayashi‡

†‡Faculty of Informatics, Kansai University.
2-1-1 Ryozenji-cho, Takatsuki-shi, Osaka 569-1095, JAPAN
{k309342,taka-k}@kansai-u.ac.jp

Abstract Nowadays, damage from a malware is increasing and cracker's purpose is changing from pleasure to profit from attacking the computer. In such a situation, it seems that malware's threat will become more serious. Hence we focus on the bot that is a kind of malware and IRC that is often used when bot communicate with C&C server. In this study, we propose the method to distinguish the command of legitimate IRC client from IRCbots by attaching an identifer to IRC client's connection. Further, we show that the activities of IRCbots can be controlled by obstructing the bot's connection using IPS.

1　はじめに

近年のインターネットの普及により，コンピュータネットワークは社会生活や生産活動において，欠かすことのできない重要なインフラストラクチャとなっている．それに伴い，悪意のあるソフトウェア（マルウェア）による被害も増加している．IPA により発表されている情報セキュリティ事象被害状況調査 [1] によるとコンピュータウィルスの遭遇率は 2002 年から 2010 年までは減少傾向で，2010 年の遭遇率は 49.1%だったが，転じて 2012 年には 71.5%と大幅に増加している．またこのようなマルウェア製作者の意図も，自分の技術力を誇示するような愉快犯的なものから，攻撃対象の情報を盗み出すことによって得られる利益を目的としたものへと変わってきている [2]．そのため，マルウェアによってもたらされる被害は昔に比べ，より深刻なものになっていると考えられる．このような状況において，マルウェアの活動を監視・抑制するような対策が必要となっている．そこ

で，本研究ではマルウェアの一種であるボットが C&C（Command and Control）サーバと通信を行うときによく利用される Internet Relay Chat（IRC）において，正規の IRC クライアントが行う通信に識別子をつけることによって，ボットが行う通信と区別し，また，ボットによる通信を侵入防止システム（IPS）を用いて遮断することでボットの活動を抑制できることを目指す．

2　ボットとは

ボットとはマルウェアの一種である．ボットに感染したコンピュータは C&C サーバと呼ばれる中継サーバを介して攻撃者（ハーダ）の指示を受け取り，スパムメールの送信・中継や複数のボットによる DDoS 攻撃，感染したコンピュータの情報収集など，様々な悪意ある活動を行うようになる．ハーダがより効率的に指示を伝えるため，一台の C&C サーバに対して複数のボットが接続していることが多く，このようにして形成される指示系統をボットネットという．C&C サーバとの通信に IRC を利用するボットを特に IRC ボットと呼ぶ．通信には，他にも HTTP や P2P が利用されることもある．ボットによる活動はわかりやすい症状が出ないため，ユーザにとって発見しづらく，またハーダによって送られてくる指示はハーダが独自に定義していることが多く，検知は困難である．荒谷らの研究 [3] では CCC Dataset2013 に含まれているボット検体の内，IRC を利用しているボットの割合の調査を行っており，約 4 割が IRC ボットであると述べている．

3　IRC とは

IRC とは TCP/IP ネットワーク上でリアルタイムでチャットを行うためのシステムのことである．ユーザは IRC 専用のクライアントソフトを用いて，ユーザ名とニックネームを設定し，IRC サーバに接続することで，同じドメインのサーバに接続している複数のユーザと会話を行うことができる．チャットのグループ単位と

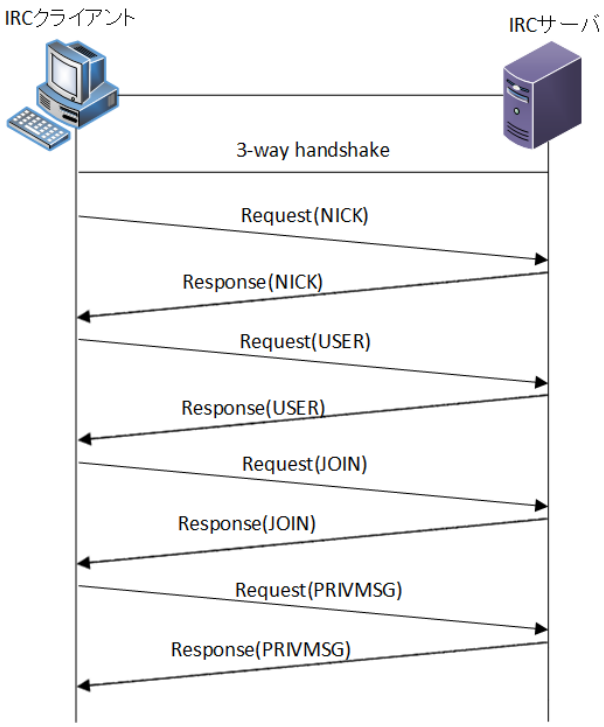

図 1: IRC クライアントと IRC サーバの通信

してチャンネルという概念があり，最初のクライアントが，ある特定のチャンネルに参加しようとした時点でそのチャンネルが作成され，そのチャンネルに参加しているクライアントの数が 0 になったとき，チャンネルは自動的に消滅する．クライアント同士の会話には，同じチャンネルに参加し，複数人で会話する方法と，相手のニックネームを指定してメッセージを送り，一対一で会話する方法がある．図 1 に IRC の一般的な利用時に行われる通信の流れを示す．NICK がニックネームの設定と変更，USER がユーザ名の設定，JOIN が指定したチャンネルへの参加，PRIVMSG が参加しているチャンネルへのメッセージ，または個人へのメッセージの送信に使われるコマンドである．

4　CCC DATASet における IRC ボットの通信挙動の調査

この章では，CCC が提供している攻撃通信データの中で，IRC ボットが C&C サーバから

指示を受け取るときに使われるコマンドの調査を行い，どのタイミングでボットであることの判断を行えば良いかを検討した．本研究では，攻撃通信データが公開されている中で最新のものである，CCC DATASet 2011 を利用した．攻撃通信データ内で利用されていた，指示伝達の方法を以下に示す．

(1) チャンネルに参加しているボットに対して PRIVMSG を利用して指示を出す．

(2) ボットのニックネームを指定して，直接 PRIVMSG で指示を出す．

(3) チャンネルのトピックにボットへの指示を設定しておき，ボットがそのチャンネルに参加したときに，自動で指示を受け取らせる．

以上のそれぞれの方法において，ボットが指示を受け取るタイミングは，(1) の方法の場合，ボットがチャンネルに参加した後，(2) の方法の場合，ボットがニックネームとユーザ名を設定した後，(3) の方法の場合，ボットがチャンネルに参加した直後となる．この三つの方法の内，最も早く指示が伝達される状態になるのは (2) の方法である．したがって，ボットは最速で，ニックネームとユーザ名を設定した時点で指示を受け取れる状態にあるといえる．

5　システムの実装

IRC ボットの活動の抑制を行うためには，IRC ボットと C&C サーバの通信を遮断する必要があるが，IRC を利用した通信そのものを禁止してしまっては，正規の IRC クライアントを利用することもできなくなってしまう．そこで本研究では正規の IRC クライアントの通信に識別子をつけることで，IRC ボットとの区別を行う．4 章の調査結果より，IRC ボットが指示を受け取れる状態にあるのは，ニックネームとユーザ名の設定後と考えられるため，識別子をつけるのは NICK コマンドを IRC サーバに送信するときのパケットとする．本研究のシステムでは，XChat[4] のモジュールに変更を加え，IRC クライアントが送信する NICK コマンドを含んだパケットの末尾に，あらかじめ conf ファイルに設定しておいた 8 文字の文字列を識別子として付け加えるようにした．XChat とは UNIX 系 OS の他にも Windows OS や Mac OS で利用できるオープンソースの IRC クライアントソフトである．

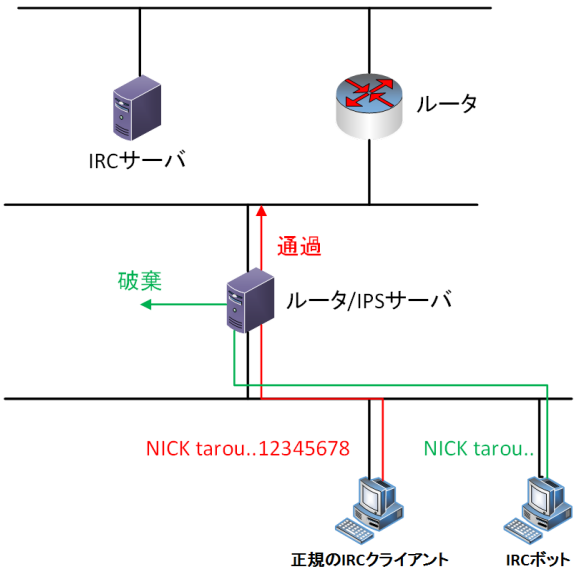

図 2: ネットワーク構成

IRC ボットによる通信の遮断には IPS を使い，ネットワーク上を流れるパケットを監視する．IPS には，IRC クライアントの NICK コマンドによるパケットで，末尾に先ほどの 8 文字の文字列がついていないパケットを発見した場合に，そのセッションを切断するように設定した．IPS の実装には，オープンソースのネットワーク型 IDS として有名である，Snort-2.9.7.0[5] を利用した．また，導入するホスト OS には全て CentOS 6.5 を利用した．本研究で構築したシステムの構成を図 2 に示す．

5.1　IDS（侵入検知システム）・IPS（侵入防止システム）

IDS とは，ネットワーク上やホスト上で通信内容を監視し，異常を発見した場合に管理者に通知を行うためのシステムである．IDS はその検知手法から，シグネチャ型 IDS とアノマリ型

IDS に大別される．前者はポートスキャンや辞書攻撃など，不正な通信を判断するためのルールをあらかじめ用意しておき，監視対象に到着したパケットとルールを照らし合わせることでその通信が悪性であるかどうかを判断し，検知を行う方式である．後者はホストの正常な状態を記録しておき，ホストになんらかの異常が発生したとき，正常状態のデータと比較することで，異常状態を判別し検知を行う方式である．

IPS は IDS にパケットフィルタリング機能を付加したシステムである．不正な通信を発見した場合，リアルタイムでその通信を遮断することが可能であり，ネットワークの監視と防御の両方の機能を備えている．本研究で使用する Snort はシグネチャ型 IDS であり，また設定により，IPS として利用することも可能である．

リスト 1: 従来の Snort のルールの例

```
alert tcp $EXTERNAL_NET any ->
$HOME_NET any (msg:"MALWARE-CNC Win.
Worm.Steckt IRCbot requesting URL
through IRC"; flow:to_client,
established; content:"JOIN |3A|#";
content:"!dl http://"; fast_pattern:
only; metadata:impact_flag red, policy
 balanced-ips drop, policy security-
ips drop, ruleset community, service
irc; reference:url,www.virustotal.com/
en/file/411
e93206a7750c8df25730349bf9756ddba52c1b
c780eaac4bba2b3872bc037/analysis/;
classtype:trojan-activity; sid:28982;
rev:1;)
```

5.2 従来の IDS・IPS を用いた IRC ボットの対策

従来の IDS のシグネチャは，IRC ボットが行う通信固有の特徴を検出することで，検知を行っている．Snort で公開されているコミュニティルールの内，IRC ボットを検知するルールを例として，リスト 1 に示す．

リスト 1 のルールは特定の文字列を含んでいるパケットを IRC ボットが受け取る指示として検知している．このようなルールは，IRC ボットの種類の特定という意味で有用であるが，一方で日々亜種が出現し，その通信挙動も変わっていくボットのすべての特徴をシグネチャで網羅することは困難である．

5.3 Snort の設定

5.3.1 IPS としての使用

Snort を IPS として運用する場合，Snort を NFQ モードで稼働させる必要がある．NFQ モードでは libnetfilter を用いてファイアウォールと連携させ，ファイアウォールから Snort にパケットを転送することで，カーネルで受け取ったパケットを Snort で取捨選択できるようにする．またこのとき，Snort はルータかブリッジに設置する必要がある．本研究ではファイアウォールには Linux に標準で実装されている iptables を利用する．Snort にパケットを転送するための，iptables の設定をリスト 2 に示す．リスト 2 で利用している NFQUEUE ターゲットは，パケットをユーザ空間に渡すための機能である．パケットは 0〜65535 のキュー番号の内，指定された番号のキューに入れることができる．本研究ではキュー番号の 2 番で Snort を待ち受けさせ，iptables は 2 番のキューにパケットを入れることで，iptables から Snort にパケットを送っている．

5.3.2 Snort のルール

Snort は NICK コマンドのパケットの内，末尾に識別子として設定した文字列がついていないパケットを破棄し，送信元に RST/ACK パケットを送信することにより，セッションを切断する．Snort に設定したルールをリスト 3 に示す．リスト 3 は 1 行目では，識別子”12345678” が付いている場合にそのパケットを通過させ，同リスト 3 行目では，識別子の付いていないパケットを破棄（drop）する，というルールセットになっ

リスト 2: iptables の設定

```
1 iptables -A FORWARD -j NFQUEUE --queue
  -num 2
```

リスト 3: Snort のルール

```
1 pass tcp 192.168.2.0/24 any ->
  EXTERNAL_NET any (msg:"IRC NICK"; sid
  :1000003; flags:PA; content:"NICK|20|"
  ; offset:0; depth:5; content:"|0d 0a
  |12345678|0d 0a|"; replace:"|0d 0a|
  AAAAAAAA|0d 0a|";)
2
3 drop tcp 192.168.2.0/24 any ->
  EXTERNAL_NET any (msg:"Bad IRC NICK";
  sid:1000004; flow:to_server,
  established; flags:PA; content:"NICK
  |20|"; offset:0; depth:5; content:"|0a
  |";)
```

ている．またこの文字列がそのまま IRC サーバに送信されてしまうと，パケットキャプチャなどにより，攻撃者に識別子を知られてしまう危険性がある．このルールでは識別子を replace 機能を使って，”AAAAAAAA” という文字列に置き換えて中継しているため，外部ネットワークからの盗聴に対して耐性を持っている．

6 評価実験

6.1 正規の IRC クライアントの検証

NICK コマンドのパケットに識別子をつけたとき，IRC サーバがそのパケットを正常に受け取り，ニックネームの設定，または変更ができるかどうか検証を行った．検証結果を表 1 に示す．

表 1 の結果から，NICK パケットの末尾に識別子をつけることは，一般に問題がないと言える．また，検証で利用した IRC サーバは，全て，NICK パケットを送信したときのレスポンスとして，”AAAAAAAA :Unknown command” という警告文を返した．このことから，IRC サーバは識別子を独立したコマンドと判断し，無効なコマンドとして無視したのだと考えられる．

6.2 IRC ボットの検証

本研究のシステムが，IRC ボットの活動を抑制できるかを検証するため，図 2 のシステムを外部から隔離したネットワーク内に構築し，入手した検体の動作実験を行った．検証には一つの検体を動作させるごとに以下の手順を踏んだ．

(1) 動作させる検体一つだけを動作環境のホストに移動させる．

(2) ファイアウォールにより，全ての通信を遮断している環境で検体を動作させ，検体が接続するドメイン名を調べる．

(3) 検体を動作させているホストの hosts ファイルを書き換え，検体を隔離ネットワーク上に立てた IRC サーバに接続するようにする．

(4) 検体が IRC サーバに接続したことを確認した後，IPS を用いて，NICK パケットの破棄を行い，その後の通信挙動の観察を行う．

(5) スナップショットを用いて，ホストを検体を動作させる前の状態に戻す．

この検証では，IRC ボットが IRC サーバに接続し，C&C サーバから指示を受け取れる状態（ニックネームの設定とユーザ名の設定の完了）にならないかどうかを，IRC ボットの活動を抑制できたかどうかの判断基準とする．検証結果を表 2 に示す．

表 2 の結果から，本研究のシステムは IRC ボットに対して有効であるといえる．

表 1: IRC サーバのニックネーム変更可否

IRC サーバ	OS	設定変更可否
ngIRCd	CentOS6.5	可
ircd-hybrid	CentOS6.5	可
InspIRCd	Windows7	可

表 2: IRC ボットの活動抑制可否

IRC ボット名	OS	抑制可否
kaiten	CentOS6.5	可
perlbot	CentOS6.5	可
sdbot	WindowsVista	可
dorkbot	WindowsVista	可
Agobot	WindowsVista	可
IRCbot	WindowsVista	可
Rbot	WindowsVista	可

7 考察

7.1 動作環境の問題

本研究で収集した検体が本来接続する C&C サーバは全て無くなっており，本研究では，ローカルネットワーク内に立てた IRC サーバを用いて疑似ボットネットを作ることで検証を行った．したがって，実際のボットネットとは多少環境に違いがある可能性もあり，実際のボットネットに IRC ボットが参加したときの活動抑制可否も検証していく必要がある．

7.2 抑制を行ったあとの IRC ボットの挙動

ほとんどの IRC ボットは NICK コマンドのパケットを破棄し，セッションを切断させると，接続元と接続先のポート番号や，接続するドメイン名を変えながら，再接続を試み続けていた．しかし，sdbot と Agobot に関しては，セッションを切断した時点で，一切の通信を行わなくなった．また Agobot に関しては，その後自身の削除を行った．このように現時点で通信妨害に対して，その後の挙動観察を妨害する仕組みが備わっている検体も存在していることが分かった．

7.3 IRC ボットが識別子を奪取する可能性

本研究では，正規の IRC クライアントが使用する識別子は平文のファイルで保存されているため，IRC ボットがそのファイルを参照すれば，その識別子を利用して，IPS のチェックをかいくぐることは十分に可能である．したがって，今後，保存している識別子の暗号化などの対策を講じる必要がある．また，IRC ボットがパケットスニファにより，内部ネットワークから正規の IRC クライアントが利用している識別子を盗聴する可能性も考えられる．このような場合も，ダミーの識別子をつけたパケットをネットワークに常時流しておくなど，なんらかの対策が必要である．

7.4 本システムの設置場所

IPS を用いたボットの検知は，ホスト単位で導入が必要なアンチウィルスソフトとは違い，ネットワーク単位での検知が可能である．しかし本システムでは識別子という外部に漏れてはならない情報を扱っている以上，どの程度の規模のネットワークセグメントごとに本システムを導入するべきなのか検討する必要がある．

参考文献

[1] 2013 年度情報セキュリティ事象被害状況調査，https://www.ipa.go.jp/files/000036465.pdf，2015 年 8 月 11 日確認．

[2] 脆弱性を利用した新たなる脅威の監視・分析による調査，http://www.ipa.go.jp/files/000017745.pdf ，2015 年 8 月 11 日確認．

[3] 荒谷　光，本間　将紘，金井　敦，斉藤　典明，”IRC プロトコルを利用した攻撃者と感染端末の探索手法”，コンピュータセキュリティシンポジウム 2013 論文集,2013(4),139-146

[4] XChat，http://xchat.org，2015 年 8 月 11 日確認．

[5] Snort，https://www.snort.org/，2015 年 8 月 11 日確認．