IRC トラフィック監視による botnet 検出システムの試作

2012SE053 池田一陽 2012SE203 長田智和

指導教員:後藤 邦夫

1 はじめに

近年 botnet ウイルスによる企業のサーバなどに対する DDoS 攻撃が問題となっている．DDoS 攻撃によりサーバがダウンし通信障害を起こして被害をだしたり, 任意で開始や停止ができるため, 脅迫に使用される場合もある．他にもスパイウェアとして botnet を利用することで大量のアカウント情報を取得することができる．その中でも DDoS の攻撃には IRC チャットサーバを踏み台にしたもののケースが多い．一般的に, 攻撃者はウイルスに感染した端末に IRC サーバのチャットを介して命令をする．この命令する IRC サーバを C&C サーバと呼ぶ. そして，感染した端末は IRC サーバを介して攻撃者から送られてきたチャットから命令を受け取り攻撃を行う．しかし botnet ウイルスを直接検出するのは難しい [0].

先行研究より, C&C セッションにおけるパケットサイズと応答時間の変化や C&C セッションにおけるプロトコルの変化がみられるため，本研究では C&C サーバと感染端末の通信の応答時間や特定のワードから botnet ウイルスに感染している端末を検出するシステムを試作する [0]. 実験方法として，Windows で動く bot の Rbot を使用する．ネットワークエミュレータの CORE(CommonOpen Research Emulator) を使用した閉じたネットワーク上で IRC サーバを構築し実験する．用意するものとして, 仮想ネットワークを作る PC と bot に感染させる PC，この 2 台の PC を使う．Rbot を動かしパケットキャプチャした結果から bot の通信の特徴を検出するプログラムを作成し，サーバ側で実際に検出が可能か実験する．

2 IRC と bot の概要

本節では IRC を利用した bot の仕組みについて説明する.

2.1 IRC と botnet

IRC とは Internet Relay Chat(インターネット・リレー・チャット) の略称である．IRC はサーバを介してクライアントとクライアントが会話をする枠組みの名称である．本来の使い方としては，多人数でのチャットや IRC クライントソフトにあるファイル送信機能などを使ってやりとりするものである．IRC チャットの一般的な会話の方法は, IRC 用のクライアント (xchat, limechat 等) を用意し, 自分のニックネームを設定して接続したい IRC サーバ名を指定してログインをする．サーバにログインしたあとは部屋と呼ばれるチャットルームの名前を入力して入る．そこで多人数とチャットを行うことができる．これらの作業は/JOIN, /server コマンドなど, コマンド操作でも可能である.

ここで使用する botnet とは悪意のあるプログラムを使用して乗っ取った多数の PC で構成されるネットワークであり，多数の PC を遠隔で指示をするために IRC サーバを利用することが多い．また，一つのサーバが停止した場合でも，別のサーバに再接続することで botnet としての機能を維持するため，止めることが難しい.

2.2 IRC を利用する bot

本来では IRCbot は利用者が在籍していない場合に自動的に会話や処理をするために存在していた．現在でも, twitter のような SNS のサービスに自動で設定したメッセージを定期的に発信したり, 特定のワードに反応してメッセージを返すものとして使用されている．ここでの bot は悪意のある攻撃機能を持ち, 他のマルウェアとの違いとしては, 自己でネットワークを構成し botnet を作る機能がある．IRC を利用する bot は Agobot，SDBot，RBot やそれ以外にいくつもの種類がある．これらにはソースコードが公開されており，開発環境も整えやすいために亜種が多数存在する [0]．これらの bot はバックドアという外部からの操作を有効にする機能を持つプログラムや DDoS Zombie という DDos 攻撃を行うための機能を持つ．他にも感染活動をする機能や自分自身を自動的にアップデートし，新機能の追加などを実行する.

2.3 IRC チャットと bot の特徴

IRC はサーバと通信するとき，TCP でポート 6665 から 6669 に接続する．特徴としては，bot に指示するとき PRIVMSG で送信される文の 1 文字目に. などの普通のチャットと指示コマンドを見分けるための文字がある．これは SDBot や他の RBot のソースコードにもあった．パケットキャプチャした結果の一部を下の表に示す.

```
50 52 49 56      PRIVMSG #test002 :.id
4d 53 47 20
23 74 65 73
74 30 30 32
20 3a 2e 69
64
```

チャットを送信, 受信するときにこの様にデータが流れる．bot は:.id のようなコマンドを処理する時，PRIVMSG #チャンネル名 :のあとにくる 1 文字目を判断し，それが指示コマンドのキーであることを確認した後，コマンドが何なのかを if 文で判断している．ソースコードでこの 1 文字目を変更することはできるが，設定することができるの

は 1byte だけなので byte 数が変わることもないと考えられる．よって:のあとにくる 1 文字目が頻出している場合などは bot の可能性があると考えられる．他にも bot 特有の応答時間の違いや，短時間のプロトコルの変化やチャネルへの単一参加が多いことである．機械的に，処理をしているので，人間的ではないことが特徴に出てきている [0]．実際にレスポンスが早いことによって平均のデータ返信時間や bot の送信データが少ないことが特徴としてある [0]．

2.4 RBot の概要

本研究で使用する RBot は，ネットワークを介した感染や IRC を介した感染など，感染方法が多く，感染が強い傾向がある．また，RBot は RxBot や様々なバージョンといった亜種が多く発見されている．

1. bot に感染した PC の OS, CPU の状態, ネットの状態を確認.
2. UDP 攻撃などの DDoS 攻撃の実行.
3. 指定したファイルのダウンロード, アップデートの実行.
4. bot 化した PC を指定したサーバ, チャンネルに接続.
5. ping 攻撃.

などがある．

RBot のソースコードは C++ で記述されている．使用した RBot は configs.h に Bot の ID, パスワード, 接続するサーバ，チャンネルを設定するだけで起動することができた．起動する際，IRC クライアントは必要としなかった．下の表は実際に IRC 上で動かした RBot の動作の一つである．

```
attacker|Hello!
attacker|.login pass1234
oqsdace |[rX ReS]: Password Accettata.
attacker|list *cmd*
attacker|.list *cmd*
oqsdace |Serching for: *cmd*
oqsdace |cmdial32.dll 04/14/2008 07:55 AM...
oqsdace |CMDIALOG.SRG 05/07/1998 00:00 AM...
...
oqsdace |Found 7 Files and 7 Directories
attacker|.id
oqsdace |[MAIN]: Bot ID: rbot01
```

RBot は感染した PC の内部にもアクセスすることができ，ハードディスクや PC の状態なども調べることができる．この実験では attacker が攻撃を指示する PC, oqsdace は RBot である．この RBot はコンパイルした後に.exe ファイルを生成し，実行すると指定した.exe の名前で動作をし，元の.exe は消えてしまう．起動をした後は，指定したサーバ，チャンネルに自動で接続を行い，ランダムな名前でログインし攻撃者の指示を待つ．表の最初の文のように普通のチャットには何も反応をしない．文の初めに. をつけたものに反応をするようになっている．最初に RBot を乗っ取るために.login とパスワードでチャットをし，成功すると成功した文が返ってくる．その後. と対応したコマンドに応じて RBot が動作する．.list コマンドは指定したファイル名が乗っ取った PC の中にあるか検索するコマンドである．他にもソースコードを見ると DDos を実行するプログラムもあった．

3 実験環境設定

本節では，実験環境の設定について記述する．

3.1 ネットワーク構成

本実験のための bot の攻撃実験を行う際，2 台の PC を閉じたネットワークで接続し実験を行う．ネットワークのイメージ図を図 1 に示す．

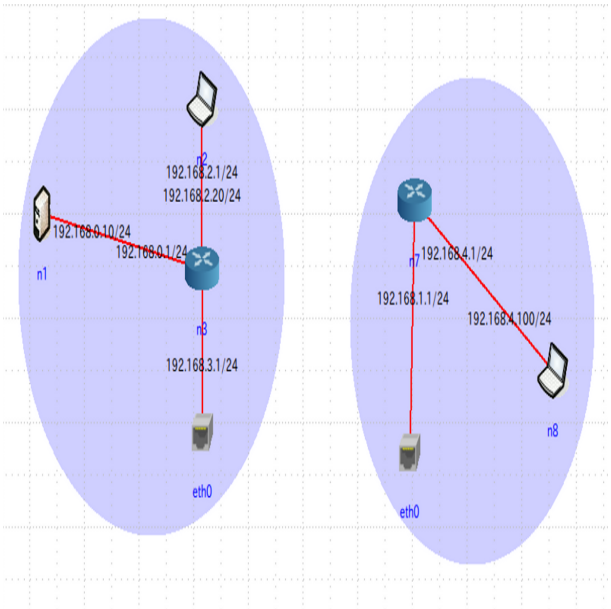

図 1 実験ネットワークイメージ図

IRC 側の IP アドレス割り当て

IRC サーバ:192.168.0.10/24
攻撃者端末 PC:192.168.2.1/24
router1(IRC 側):192.168.0.1/24
router1(攻撃者端末側):192.168.2.20/24
router1(eth0 側):192.168.3.1/24
Core の eth0 と実世界の eth0 のブリッジ:192.168.1.20/24

windowsXP 側の IP アドレス割り当て

VM 内の windowsXP:192.168.4.100/24
router2(eth0 側):192.168.1.1/24
router2(VM 側):192.168.4.1/24
VM の Vboxnet:192.168.4.210/24
eth0:192.168.1.100/24

IRC サーバには ircd-hybrid7.2 を使う．RBot は windowsOS 向けに開発されたものであり，セキュリティ面で動作のしやすく，virtualbox で動作しても軽かった windowsXP を使用する．botPC は linux14.04 の virtualbox 内の仮想環境で windowsXP を使用し，RBot を感染させる．attackerPC(攻撃者) の OS は linux10.04, IRC クライアントは xChat を使用する．IRC サーバ側で wireshark を使用し，通信を確認する．

3.2 CORE

オープンソースな仮想ネットワークエミュレータである．主な特徴としては，GUI(Graphical User Interface) 操作で使いやすく，軽量の仮想マシンである．CORE には仮想ネットワーク上にある通信路と他のネットワークインターフェースを接続できる RJ45 というノードがある．本実験ネットワークの構築の際には RJ45 を eth0 と接続し，スイッチングハブを通して外部と通信できるようにする．RJ45 は eth0 と接続をし，起動を行うと自動でブリッジが作られ，eth0 と veth が接続される．これを解決するには手動で IP アドレスをつける必要がある．

3.3 IRC 側のネットワーク設定

実験には core を使用した．core 内の host に ircd-hybrid7.2 を使い irc サーバを構築する．IRChost, attackerPC に 192.168.2.1/24 で IP アドレスをつける．二つをルータに繋げ，ルータを RJ45 に接続する．RJ45 は core の機能の一つであり，実世界の eth0 と core を接続が可能となる．RJ45 は core を起動している linux10.04 の eth0 に接続する．しかしこのままでは core によって生成されたブリッジによって eth0 が外部と接続できないため，ifconfig を使い手動でブリッジに IP アドレスをつける．

3.4 WindowsXP 環境設定

VirtualBox の設定は，環境設定のホストオンリーネットワークを作り，IP アドレスは 192.168.4.210/24 とする．そして，VM(WindwsXP) の設定は，ファイヤーウォールを無効にして, 固定の IP192.168.4.100/24 を割り振る．そして Default gateway を 192.168.4.1/24 に設定する．そして，ホストオンリーアダプターに設定する．ホストオンリーアダプターだと，vboxnet0 というインターフェースに繋がる．このように，VM がどのように繋がっているのかがわかりやすいため，このアダプターにする．そして，ここから eth0 に繋ぐところで，うまくつながらなかったので，vboxnet0 と eth0 の間に core をつないで，ルータにする．スイッチングハブを用意してそれぞれの PC の eth0 を接続し Ping を飛ばしたところ接続が確認された．こうすることで IRC までの接続が可能となった．

3.5 Bot の動作実験

入手した bot は SDbot，RBot，RxBot の 3 種だったが，SDbot，RxBot は設定をしても動かないものや，パスワードが認証されずに操作ができなかったために使用できなかった．RBot は設定通りに動いたため今回の実験には RBot を使用する．実際に IRC サーバを構築し，RBot のソースコードに IRC サーバの IP アドレスを記述する．チャンネルはここでは test002 で実験をした．ソースをコンパイルすると rbot.exe が生成された．それを実行すると，ソースコードで記述した bottest.exe が起動した．起動した時点で RBot は指定した IRC サーバ，チャンネルに接続をし，ランダムでつけられた名前で待機をした．linux10.04 の xchat で同じサーバ，チャンネルに接続をし，実際にコマンドを打ってみたら返答があったため RBot の動作実験は成功した．

3.6 システムの概要

IRC を利用した bot の指示は指定した IRC サーバ, チャンネル内で行う．攻撃コマンドのやりとりは PRIVMSG という一人のユーザが全体に向けて送信するチャットで行う．そのチャットの 1 文字目に bot の config ファイルで指定した 1byte の文字の後の文字で判別する．図 2 は IRC

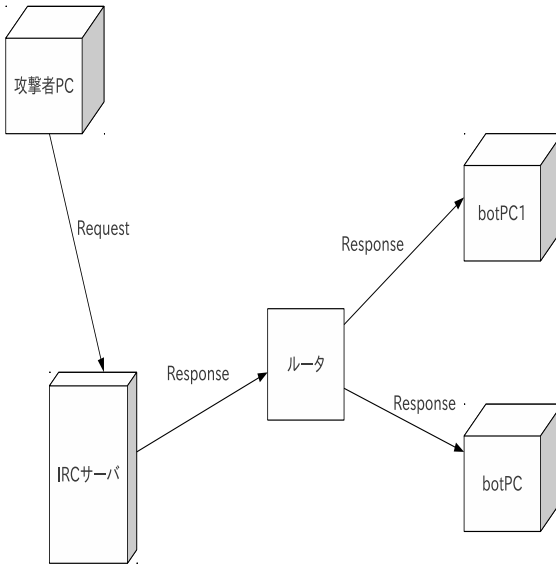

図 2 IRC チャット送信の流れ

チャットを送信した際にチャットがどのように流れていくか表した図である．攻撃者 PC から出した指示は Request という形で IRC サーバを経由して botPC にそれぞれ送信される．2 節の IRC の特徴の実験を例にすると, Request, Response の送信データには PRIVMSG test002 :.id　が送信される．これより, 本システムはこの攻撃コマンドの

ための 1 文字目を判別するプログラムを作成した．このプログラムは bot 側の外部と接続するルータで動作する．プログラムは外部に接続するルータに流れてくるデータをパケットキャプチャをし，IRC サーバが設定で開いているポート 6665 から 6669 に送信する通信のみを抜きだし，PRIVMSG #チャンネル名 :の後にある 1 文字目を判別しポート毎にポート名の名前のついた.txt ファイルを生成し 1 文字目のみを書き込んでいく．1 文字目が連続して出た場合は連続して書き込み，一定数を越えたら警告を出す．連続して出なかった場合は最後に出た 1 文字目を新しく上書きをして保存する．

:後の 1 文字判別のソースコード

```
if( ( sp = strstr(tmpstr, ”PRIVMSG”) ) != NULL
)
char *split1, *split2;
split1 = strtok(sp, ”:”);
split2 = strtok(NULL, ”:”);

if(split2 != NULL)
char ssp[1024], ssp2[1];
sprintf(ssp, ”
```

4 実験

本節は，実験の手順と実験の結果について記述する．

4.1 実験手順

1. 3 節のネットワークを構築する．本システムを bot 側の外部接続を行うルータで起動する．IRC サーバの IP アドレス 192.168.0.10/24 と実験で使用するチャンネル test002 を RBot の config ファイルに設定する．
2. RBot を起動する． linux10.04 側の攻撃者用端末も /server, /join コマンドを使用して同じ IRC サーバ, チャンネルに接続する．
3. 攻撃者端末から図 3 のようにいくつか RBot に指示コマンドを出し，プログラム側でエラーが出ているか確認する．
4. 同じネットワークで RBot の接続ポートを変更しても検出できるか，普通のチャットを行って誤検出がないかを再度実験する．

4.2 結果

図 3 で行った実験では bot の通信を検出することはできた．実際に下のようにプログラムでエラーが発生した．

```
Warning!! Attack command from Port: 55458!!
```

また，RBot のポートを変更した場合でも同様に検出することができた．他にも，RBot を使わずに普通の IRC クライアントを複数用意し，実際にあった会話をしてみたが警告文がでることはなかった．これは同じ人物が連続で文の初めに同じ文字を 5 回使わなかったから警告文がでなかったためであり，偶然警告が出て後検知する可能性があると考えられる．しかし，bot 側の端末では PRIVMASG が攻撃者端末からの指示の返答によるものなので，感染端末の検出はできなかった．

```
      * 今 #test002 で会話中です。
attacker .login pass1234
zolurtve [rX ReS]: Password Accettata.
attacker .list *cmd*
zolurtve Searching for: *cmd*
zolurtve Found 0 Files and 0 Directories
attacker .driveinfo
zolurtve [Rx DoS]: Disk Drive (C:¥): 31,447,204KB Disko, 21,791,752KB Libero,
zolurtve [Rx DoS]: Cdrom Drive (D:¥): Non Trovato.
zolurtve [Rx DoS]: Cdrom Drive (E:¥): 627,628KB Disko, 0KB Libero, 0KB .
attacker .findpass
attacker .id
zolurtve [MAIN]: Bot ID: rbot01.
attacker .nick newbotname
      * zolurtve は newbotname にニックを変更しました。
```

図 3 IRC 実験

5 おわりに

本研究ではサーバにくる通信の PRIVMSG から攻撃者端末，踏み台に使われている C&C サーバの検知は成功した．また，今回実験で使用できなかった SDBot，RxBot もソースコードを見るかぎりではコマンドの 1 文字前に 1byte 文字を設定するので，同じシステムで検出が可能と考えられる．特定の bot を検出する場合には使用するコマンドを検出するようにすれば検出率もかなり上がると考えられる．本研究では攻撃者端末，C&C サーバは検知できたため，そこから感染端末の検出も間接的だが可能である．さらに条件を絞り込み，監視を強化することで感染端末の特定の可能性が上がる．また，本研究のシステムでは感染端末の検知はできなかった．検知の方法の一つのレスポンスタイムの機能を導入することで普通のチャットとの誤検知率の低下と感染端末の割り出しも可能と考えられる [0].

参考文献

[1] JPCERT: ボットネットの概要 (accessed Jul. 2006). https://www.jpcert.or.jp/research/2006/Botnet_summary_0720.pdf.

[2] 阿部 義徳, 田中 英彦 : C&C セッション分類によるボットネットの検出方法の一検討，情報理工科学技術フォーラム一般講演論文集 (2007).

[3] 勝村 幸博 : インターネット上の新たな脅威「ボット (bot)」に気をつけろ！ (accessed Dec. 2004). http://itpro.nikkeibp.co.jp/free/ITPro/OPINION/20041215/153889/.

[4] 星澤 裕二 : レスポンス・タイムによるボット検知手法 (accessed Jul. 2006). http://itpro.nikkeibp.co.jp/article/Watcher/20060729/244644/.