Vol.2018-IOT-42 No.3
2018/6/28

悪性 Botnet 包囲網における P2P 通信検知の試み

山之上卓[†1]

概要：悪性 botnet の P2P 通信を検知しようとする悪性 botnet 包囲網(良性 botnet)について述べる．P2P 通信のような botnet の技術に対応するため，以前から開発している我々の良性 bot を利用して，悪性 botnet 包囲網(良性 botnet)を開発している．良性 botnet は Agent bot と Analyzing bot の２種類の良性 bot のグループである．悪性 botnet の P2P 通信を 1 台の IDS で検知することは難しいが，我々の良性 botnet は複数の良性 bot を協調動作させることにより，P2P 通信を検知する能力を持つ．この良性 botnet により，悪性 botnet の通信を真似する偽 botnet の通信を検知することができた．

キーワード：Bonet, Bot, Wiki, P2P, IDS, 分散協調,

An Attempt to Detect P2P Communication by the Malicious Botnet Capturing Network

TAKASHI YAMANOUE[†1]

Abstract: A malicious botnet capturing network (beneficial botnet), which tries to cope with peer to peer (P2P) networking of malicious botnets, is discussed. In order to cope with such botnets' technology, we are developing a beneficial botnet as an anti-bot measure, using our previous beneficial bot. The beneficial botnet is a group of beneficial bots which are Agent bots and an Analyzing bot. A P2P communication of malicious botnet is hard to detect by a single Intrusion Detection System (IDS). Our beneficial botnet has the ability to detect P2P communication, using collaboration of our beneficial bots. The beneficial bot could detect communication of the pseudo botnet which mimics malicious botnet communication.

Keywords: Botnet, Bot, Wiki, P2P, IDS, Distributed, collaboration

1. はじめに

多くのネットワーク管理者やセキュリティ担当者が悪性 Botnet に頭を抱えている．悪性 Botnet は多くの人々に様々な場所から迷惑メールをばらまいたり，その超並列特性を使って，多くのパケットを短時間に 1 台のサーバに送信することにより，DDoS 攻撃を行ったりする．Botnet の Bot はゾンビコンピュータの利用者の銀行口座を盗むことも行う．Botnet は粘り強い性質も持つ．情報セキュリティ担当者が Botnet の中のいくつかの Bot を見つけて対処しても，Botnet は悪事を続けることができる．悪性 Botnet は情報セキュリティ担当者の bot 対策を潜り抜けるため，日々進化している．

2000 年代中頃に現れた Agobot/Phatbot [6]などの Botnet には bot 対策を潜り抜けるため Peer To Peer (P2P)ネットワークの技術を使っている．2000 年代後半に現れた conficker[2]などの Botnet はドメイン生成アルゴリズム (Domain Generation Algorithm, DGA)の技術を使っている．

Gameover ZeuS は有名な botnet である．これは警察組織の国際的な連携により，2014 年に壊滅したが，FBI の公表[4]によると，損失は 1 億ドルに上ると見積もられている．Andriesse と Bos によると，「２番目の中央集中型の Zeus は P2P Zeus または Gameover として知られる peer-to-peer (P2P)の変種に変異した．P2P Zeus は中央集中型 command and control (C2) サーバには依存しないため，従来型の Zeus に対する対応に免疫がある」[1]．このように，P2P 機能を持った botnet は検知しにくく，壊滅させるのが難しい．Gameover Zeus が壊滅までにこのような多額の損失を与え，国際的な連携のような大きな努力を必要としたのは Gameover Zeus が P2P 機能を獲得したことも原因の一つと考えることができる．Gameover ZeuS は壊滅したが，同様の P2P botnet が組織内で活動する可能性はある．

組織の出入り口に設置された 1 台の侵入検知システム (Intrusion Detection System, IDS) で組織内のボットの P2P 通信を検知することは難しい．

このような Botnet の技術に対処するため，我々が従来から開発を続けていた良性 Bot[13][14][15][16][17]を使って，悪性 Botnet 包囲網(良性 Botnet) を開発している．良性 Botnet は LAN の Nat の内側に設置する Agent Bot と，Agent Bot によって選択収集されたデータを解析する Analyzing Bot により構成された，良性 Bot のグループである．我々の良性 Botnet は，良性 Bot に協調動作をさせることにより，P2P 通信を検出する能力を持つ．

†1 福山大学
Fukuyama University

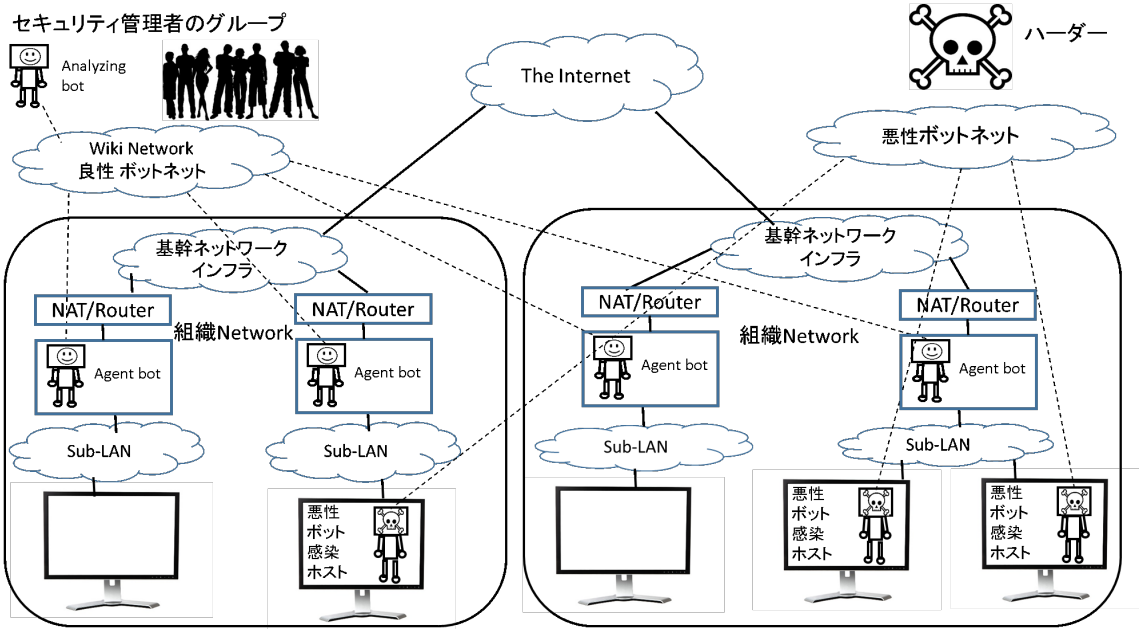

図 1　悪性 Botnet 包囲網(beneficial botnet)の概要

Figure 1　Outline of the beneficial botnet

この能力を検証するため，我々は P2P 通信を行い，それ以外は悪影響を与えない簡単な Botnet (偽 GameOver ZeuS) を作成した．

複数の Nat で防御された LAN を含むネットワークと良性 Botnet を組み合わせた実験ネットワークを構築し，その実験ネットワークで偽 GameOver Zeus を動作させると，良性 Botnet により 偽 GameOver Zeus の通信を検知することができた．

2. 悪性 Botnet 包囲網

組織のネットワークの出入り口に IDS や IPS を設置して利用することが良く行われている．Communication and Control (C2)サーバとすべての bot の間の通信がそれによって検知可能であるので，組織の出入り口に設置された IDS や IPS は中央集中的な Communication and Control (C2)サーバを持つ悪性 Botnet には有効である．

しかしながら，IDS や IPS の内側の，組織内で行われる P2P 通信をこのような IDS や IPS で検知することは困難である．我々は，Gameover ZeuS のような P2P 通信機能を持つ botnet に対処する悪性 botnet 包囲網(beneficial botnet)を設計している．

2.1 悪性 bonet 包囲網の概要

この論文で述べる悪性 botnet 包囲網(beneficial botnet) は，従来の IDS や IPS の欠点を克服しようとするものである．図 1 に悪性 botnet 包囲網の概要を示す．Agent bot は Sub-LAN と NAT または Router の間に設置される．Agent bot は Sub-LAN と Sub-LAN の外部との通信データを収集する．Analyzing bot は Agent bot が収集したデータを集めて解析し，sub-LAN 内に潜む bot を検出する．すべての Agent bot と Analyzing bot はインターネット上の Wiki のページに書かれた script によって制御される．

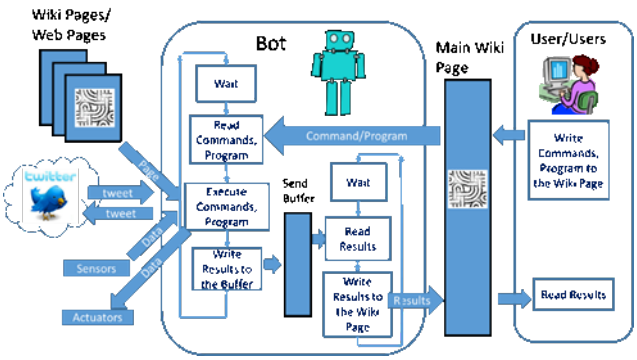

図 2　Bot の振る舞い

Figure 2　Behavior of a bot

2.2 悪性 bonet 包囲網の bot

Beneficial botnet で利用する beneficial bot は wiki ソフトウェアの wiki ページ上に書かれたスクリプトのインタープリタである.図 2 に beneficial bot の振る舞いを示す．これは以下を繰り返す．

1) 一定時間停止する．
2) コマンドとプログラムで構成されたスクリプトを，事前に bot に指示されていた wiki ページ(main wiki)から読み込む．
3) コマンドとプログラムを実行する．プログラムは main wiki の他に，他の wiki ページや web ページを読み込むことができる．もし，bot が agent bot であるなら，コマンドによって，bot が動いているコンピュータのネットワークインターフェース間の通信データを

収集することができる．Agent bot は，インターフェース間の通信をコマンドによって制御することもできる．もし，bot が analyzing bot であるなら，R 統計計算システムを用いて，収集されたデータを解析することができる．

4) Bot の実行結果は送信バッファに書きこまれる．送信バッファに書きこまれたデータは，スクリプトが書かれた wiki ページに書きこまれる．

図 3 に script を書いた wiki ページの例を示す．result: の行より前の部分が script のコマンドとプログラムであり，result: より後の部分が実行結果である．

```
objectPage http://www.[redacted]
device yamaRasPiDp9_1 or yamaRasPiDp9_2 start after no w
command: set readInterval=60000
command: set execInterval=0
command: clear sendBuffer;
command: program ex1
program: s=0;
program: for i=0 to 10
program:   s=s+i
program:   ex("service","putSendBuffer "+s)
program: next i
command: end ex1
command: run ex1
command: ex("service","sendResults.")
result:
0
1
3
6
10
15
21
28
36
45
55
currentDevice="yamaRasPiDp9_1",Date=2018/5/15/ 12:54:17
```

図 3　Bot が実行する Script の例

Figure 3　Example of the Script which is interpreted by a Bot

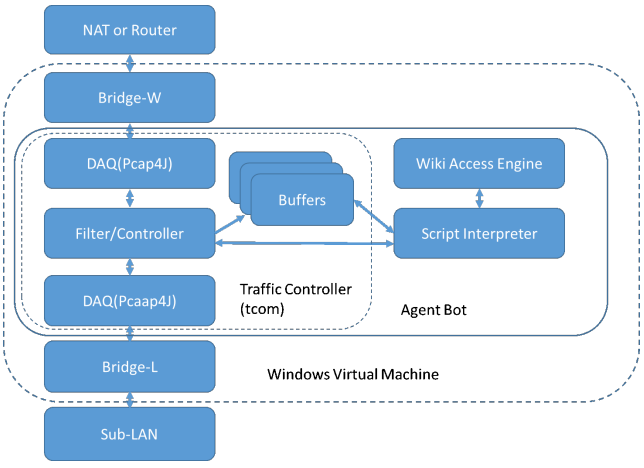

図 4　Agent Bot の構成

Figure 4　Structure of the Agent Bot

コマンドは command: で始まる行に書かれていて，プログラムは program: から始まる行に書かれている．この Script は，1 から 10 までの和を計算し，その途中経過も含めて，result: より後に書きこんでいる．

コマンドとして，“set pageName <page-name>” と “include <url>” も使うことができる．“set pageName <page-name>” が bot のインタープリタで解釈されたとき，bot は次に読み込む main wiki ページを，元の main wiki ページと同じ wiki の，<page-name>で示されたページで置き換える．<page-name>の一部としてその時の時間や日付を使うこともできる．これにより，日時に応じた別のページに実行結果を書きこむことが可能になる．“include <url>” が bot のインタープリタで解釈されたとき，bot はその部分に <url> で示された wiki に書かれている script を埋め込む．これにより，オブジェクト指向プログラミングにおいて複数のオブジェクトが 1 つの class で書かれたプログラムを実行するのと同様に，複数の bot が同じ script を実行することが可能になる．Include される script を含んだページを class ページと呼ぶことにする．それに対して，main wiki のことを object ページとも呼ぶことにする．

Bot は http を使って wiki ページと通信しているため，Bot が NAT で守られた LAN の中にいた場合でも，main wiki のページをインターネットや Bot から見える場所にある Wiki のサーバに置くことにより，Bot の管理者は Bot を NAT の外部から制御することが可能になる．

2.3 Agent Bot

Agent bot は”Traffic controller” (tcon) を備えた良性 bot である．

Tcon は 2 つの network interfaces とデータ取得ライブラリ(DAQ)と Filter/Controller と複数の Buffer を持っている．DAQ として Pcap4j を使っている．

この Bot の script interpreter は tcon の操作も行う．Bot は 2 つの interface の間の通信データを収集し，main wiki の script によってそのデータを main wiki のページに書きこむ．2 つの interfaces 間の通信の制御することもできる．Network interface の一つを NAT または router に接続し，もう一つを sub-LAN に接続することにより，sub-LAN と sub-LAN の外との間の通信のすべてを収集し，制御することができる．図 4 に Agent bot の構造を示す．

tcom は以下の buffer を持つ．通信データはその種類によって分類され，これらの buffer に格納される．

- Packet-history

この buffer は 2 つの interface 間で転送されるパケットの情報を格納する．この buffer は同じ source IP address と destination IP address のペア同士を格納するための sub buffer を持っている．Packet の payload の sha1 hash とその

packet が転送された時間と packet の情報もペアと一緒に格納される. Sha1 の値は 2 つの異なる sub-LAN 間で行われる P2P 通信を検知するために使われる.

Sub buffer があふれた時, 古い情報が削除される.

- MAC-list

この buffer は sub-LAN 内でやりとりされるすべての frame の MAC address を格納する. MAC address に対応付けられたすべての IP address も Mac address に関連付けて格納される. MAC list はこの sub-LAN に接続されているホストや, LAN 内の IP address を知るのに使うことができる. 他よりも多くの IP アドレスが結び付けられている MAC address を見つけることにより, default gateway を特定することもできる.

- Domain-list

この buffer は sub-LAN 内の DNS query をその query が行われた時間と一緒に格納する. Domain-list は sub-LAN のどのホストが LAN 外のどのホストと通信したか知るために使うことができる. この list は DGA の利用を検知するためにも利用することができる.

- Dhcp-list

この buffer は DHCP 通信の結果をそれが行われた時間と共に格納する. この list は DHCP spoofing や不正 DHCP サーバの検知を行うのに使うことができる. Default gateway を特定するのにも役立つ.

- Arp-list

この buffer は Arp 通信をそれが行われた時間と共に格納する. Arp spoofing の検知に使うことができる.

Agent bot はこれらの buffer から情報を得たり, 操作したりするコマンドを実行できる. 2 つのネットワーク間の通信を制御するコマンドも持っていて, このコマンドを使うことで悪性 bot の通信を見つけた時, その通信を遮断することもできる.

2.4 Analyzing Bot

Analyzing bot は各 agent bot で収集された情報を集め, それを解析する.

すべての良性 bot の言語プロセッサは, comma separated value (CSV) parser と表操作/表計算関数を備えている.

Analyzing bot はこれに加えて R 統計計算システム[5]も備えている. Analyzing bot の script の中に R のプログラムを埋め込むことができる. 良性 bot が元々備えている言語プロセッサと R 統計計算システムの間で値を交換する関数も使うことができる.

3. 実験

我々は beneficial botnet を使って実験的な bot 検知基盤を実装し評価した. この基盤の有用性を評価するため, 悪事を行わずに, bot 間の P2P 通信を行う偽 Gameover ZeuS の実装も行った.

3.1 偽 Gameover ZeuS

Gameover ZeuS は Command and Control (C2) server と Web サーバである C2 Proxy と P2P ネットワークのノードである Harvester bot によって構成されている. Harvester bot の一部は proxy bot として, C2 Proxy と P2P ネットワークの間の通信を担う(図 5).

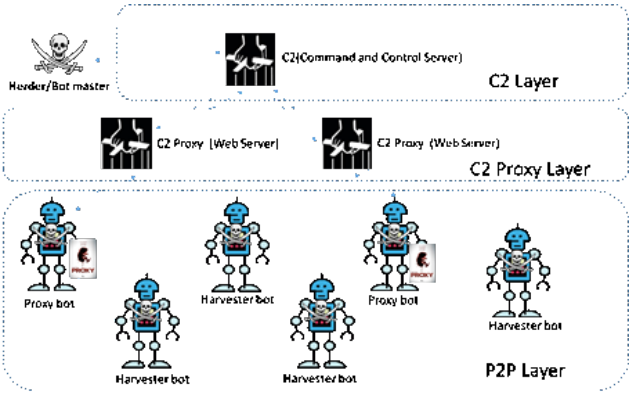

図 5 Gameover ZeuS の構成

Figure 5 Structure of the Gameover ZeuS

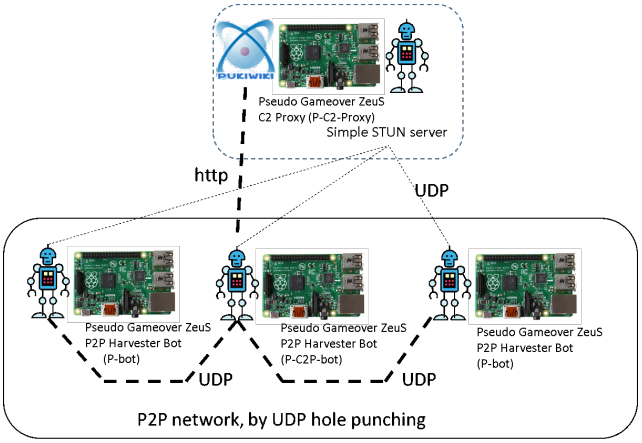

図 6 偽 Gameover ZeuS

Figure 6 Pseudo Gameover ZeuS

偽 Gameover ZeuS は, C2 Proxy に相当するホスト(P-C2-Proxy)の Wiki ページに書かれたコマンドを, Proxy bot に相当する P2P Harvester bot (P-C2P-bot)が読み込み, それを, P2P ネットワークを通じて, Harvester bot に相当する, 偽 Gameover ZeuS の他の P2P Harvester bot (P-bot)に転送する. P-bot はコマンドを読んでもなにも実行しない. 偽 Gameover ZeuS の P2P ネットワークは, UDP ホールパンチ

ングを使って, bot 間を UDP 通信で結んでいる. UDP ホールパンチングを行う為, P-C2-Proxy で簡易 STUN サーバを動かしている(図 6).

偽 Gameover ZeuS は良性 bot に P2P 通信の機能を加えるなどして作成した.

実験では, P-C2-Proxy に書かれたコマンドを, P-C2P-bot が 30 秒に一回読み込んで, それを P2P ネットワークの, 他の P-bot に転送させた. P-bot および P-C2P-bot 間は, UDP ホールを維持するため, 20 秒に一回, keep alive とその ack パケットのやりとりを行わせた.

3.2 実験ネットワーク

Beneficial botnet が P2P 通信の検知を行うことができるか否かを知るために, 図 7 で示す実験のための bot 検知基盤に偽 Gameover ZeuS を配置したネットワークを構築し, 偽 Gameover ZeuS と beneficial botnet を動作させた.

3.3 Agent Bots の `script` と実行結果

図 8 に P2P 通信を検知するための, agent bot の script を示す. この script は, 1 分以上繰り返されている同じ (source IP, destination IP)のペアの unicast 通信の中で, UDP プロトコルで, source Port および destination Port それぞれが, 123 (NTP)ではなく, かつ, 53 (DNS)ではないもののパケットの情報を集めることを表している. 図 9 に Agent bot が収集したデータが書きこまれた, Agent bot の object page の一部を示す. 図 8 の script は図 9 の include コマンドにより読み込まれている.

以下に, 偽 Gameover ZeuS の P2P 通信をとらえた出力行の 1 行の例を表す.

cmd=get repeating, date="2018/04/14 17:03:19 +0900", no=3299, if=1, smac="bc:5c:4c:5d:1c:cd", dmac="b8:27:eb:cb:d6:38", prtcl=udp, sip="192.168.13.160", dip="192.168.2.100", sp=34724, dp=33331, sha1payload="9dac7a7beb944a7193847a3d0fbcc370d13a5838", payloadLength=46, payload=broadcast id=3394824 ttl=3 cmd="message test".....

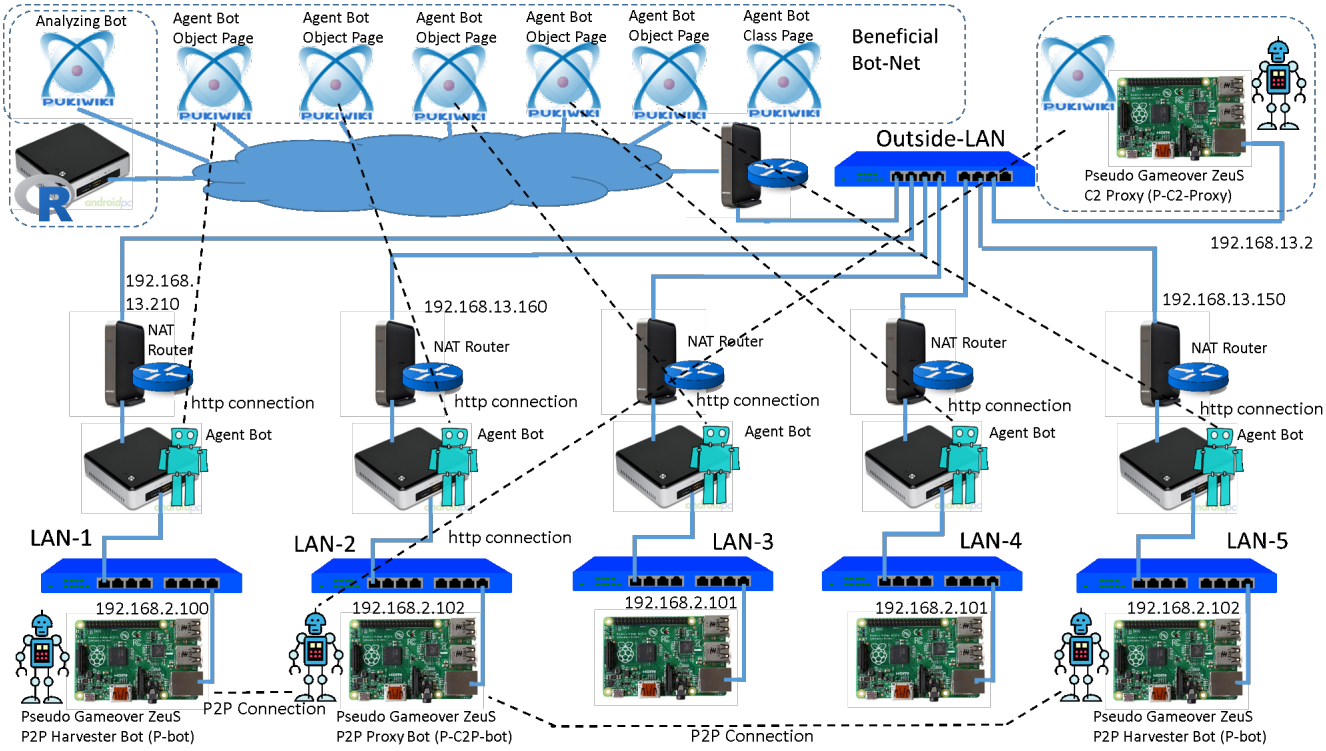

図 7 Bot 検知基盤に偽 Gameover ZeuS を配置した実験ネットワーク

Figure 7 Experimental network consists of the beneficial botnet and the Pseudo Gameover ZeuS

図 9 で示した Agent bot の Object page には以下に示す, LAN-2 と LAN-5 の間でパケットが移動したことを示す行も含まれていた.

lan1= 1 ,date= 2018/04/14 17:03:18 +0900 ,smac= b8:27:eb:2f:33:cd ,dmac= bc:5c:4c:5d:1a:c9 ,sip= 192.168.2.102 ,dip= 192.168.13.150 , lan2= 4 ,date= 2018/04/14 17:03:17 +0900 ,smac= bc:5c:4c:5d:1a:bf ,dmac= b8:27:eb:3a:6b:fa ,sip= 192.168.13.160 ,dip= 192.168.2.102 ,sha1payload= 9dac7a7beb944a7193847a3d0fbcc370d13a5838 ,payload = broadcast id

その Object page には LAN-3 と LAN-4 に出入りする P2P 通信の存在を示す行はなかった.

図 8 の script は host list と domain list も出力する. Host list はどの位の, どのようなホストが sub-LAN に接続されているかを知るために使うことができる. ルータや NAT の場合は, 1 つの MAC address に対して多くの IP アドレスが結び付けられるので, ルータや NAT がどのホストかを知ることもできる.

```
command: set readInterval=180000
command: set sendInterval=180000
command: set execInterval=0
command: tcon clear all.
command: set reportLength=600
command: clear sendBuffer.
#command: tcon get repeating unicast over 300000.
command: program ex1
program:  ex("tcon","set repeating number=20.")
program:  output10=ex("tcon","get repeating unicast over 60000.")
program:  output11=grep(output10,"prtcl=udp")
program:  output11=grepNot(output11,"dp=123,")
program:  output11=grepNot(output11,"sp=123,")
program:  output11=grepNot(output11,"dp=53,")
program:  output11=grepNot(output11,"sp=53,")
program:  ex("service","print ln "+output11)
program:  ex("service","putSendBuffer "+output11)
program:  output20=ex("tcon", "get hosts.")
program:  ex("service","print ln "+output20)
program:  ex("service","putSendBuffer "+output20)
program:  output30=ex("tcon", "get domain-list.")
program:  ex("service","print ln "+output30)
program:  ex("service","putSendBuffer "+output30)
#command: tcon get dhcp-list.
#command: tcon get arp-list.
command: end ex1
command: clear sendBuffer
command: run ex1
command: sendResults.
```

図 8 すべての Agent bot に対する指示を行う Class page の Script

Figure 8 Script of the Class page to direct all of agent bots

図 9 1 つの Agent bot の Object page の一部

Figure 9 A part of the object page for an agent bot

悪性 Bot が DGA を使うとき, 定期的に, 短時間に多くの不審な FQDN の問い合わせを DNS に行い, その問い合わせの大半は失敗するので(Gameover ZeuS の場合), Domain list は DGA を検知するにも役立つ場合があり得る.

3.4 Analyzing Bot の script と解析結果

Analyzing bot は異なる sub-LAN で動作している Agent bot が集めたパケットのデータを読み込んで, 異なる 2 つの sub-LAN で、payload の hash 値が同じパケットが短い時間内に出現するか否かを調べる. なお, Analyzing bot では UDP の unicast で, なおかつ, NTP や DNS ではないパケットの情報のみ収集している.

図 10 に Analyzing bot の script の一部を示す.

R: で始まる行が R 統計計算システムのプログラムである. この中で, 右端に::がある部分は, R の関数の定義が次の行に続いていることを表す. 図 10 の中で定義されている R の関数の p2pPacketList は, Agent bot で収集されたデータの中で, 異なる 2 つの sub-LAN の packet の中で同じ payload (同じ sha1)を持ち, 一定時間内に収集されたものを取り出して, 2 つの sub-LAN におけるパケットの情報を 1 つのベクトルにまとめている.

以下は, 実験を行った時に, Analyzing bot の object page に書かれた結果の 1 行を示す. この行は, 図 11 で示すように, パケットが LAN1 から LAN 2 に流れた可能性が高いことを示している. lan1=0 は LAN1, lan2=1 は LAN2 を表す.

Part(a)

```
command: set readInterval=300000
command: set execInterval=0
#command: set sendInterval=300000
command: set sendInterval=0
command: R ex1
R: print("start")
R: rm(list=ls())
R: p2pPacketList<-function(vec){                                        ;;
R:   p2pList <- as.numeric(NULL)                                        ;;
R:   vecLength <- nrow(vec)                                             ;;
R:   for(i in 1:(vecLength-1)) {                                        ;;
R:      for (j in (i+1):vecLength) {                                    ;;
R:         x<-vec[i,]                                                   ;;
R:         y<-vec[j,]                                                   ;;
R:         if(!is.null(x) && !is.null(y) && (x$SHA1 == y$SHA1)) {        ;;
R:            if(x$Node != y$Node)                                      ;;
R:            if(as.numeric(abs(as.POSIXct(x$Date)-as.POSIXct(y$Date)), ;;
R:                          units="secs")< 5) {                         ;;
R:               aLine <- paste("lan1=", x$Node,                        ;;
R:                              ",date=",x$Date,                        ;;
R:                              ",smac=", x$SMAC,                       ;;
R:                              ",dmac=",x$DMAC,                        ;;
R:                              ",sip=",  x$SIP,                        ;;
R:                              ",dip=", x$DIP,                         ;;
R:                              ", lan2=",  y$Node,                     ;;
R:                              ",date=",y$Date,                        ;;
R:                              ",smac=", y$SMAC,                       ;;
R:                              ",dmac=",y$DMAC,                        ;;
R:                              ",sip=",  y$SIP,                        ;;
R:                              ",dip=", y$DIP,                         ;;
R:                              ",sha1payload=",  y$SHA1,               ;;
R:                              ",payload=", y$PayLoad )                ;;
R:                p2pList <- append(p2pList,aLine)                      ;;
R:            }                                                         ;;
R:         }                                                            ;;
R:      }                                                               ;;
R:   }                                                                  ;;
R:   return (p2pList)                                                   ;;
R: }
command: end ex1
```

図 10　Analyzing Bot の Script の一部

Figure 10　A part of the Script of the Analyzing Bot

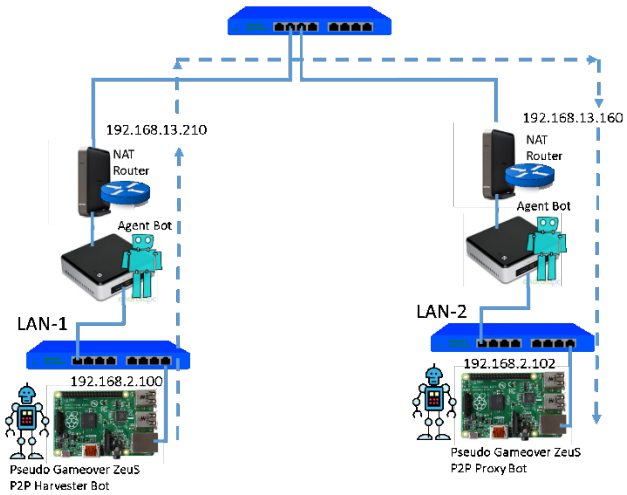

図 11　“lan1= 0 ,⋯ ,sip= 192.168.2.100 ,dip= 192.168.13.160 , lan2= 1 ,⋯ ,sip= 192.168.13.210 ,dip= 192.168.2.102 ,⋯”がその可能性を表すパケットの移動

Figure 11　Path of the packet which is estimated by the “lan1= 0 ,… ,sip= 192.168.2.100 ,dip= 192.168.13.160 , lan2= 1 ,… ,sip= 192.168.13.210 ,dip= 192.168.2.102 ,…”

lan1= 0 ,date= 2018/04/14 17:06:50 +0900 ,smac= b8:27:eb:cb:d6:38 ,dmac= bc:5c:4c:5d:1c:cd ,sip= 192.168.2.100 ,dip= 192.168.13.160 , lan2= 1 ,date= 2018/04/14 17:06:51 +0900 ,smac= bc:5c:4c:5d:1a:c9 ,dmac= b8:27:eb:2f:33:cd ,sip= 192.168.13.210 ,dip= 192.168.2.102 ,sha1payload= f14db4dae7a139cde5185267b8d353498850f22b ,payload = broadcast id

object page の結果部分には以下の行も含まれていた．これは LAN-2 と LAN-5 の間でパケットが移動した可能性が高いことを表す．

lan1= 1 ,date= 2018/04/14 17:03:18 +0900 ,smac= b8:27:eb:2f:33:cd ,dmac= bc:5c:4c:5d:1a:c9 ,sip= 192.168.2.102 ,dip= 192.168.13.150 , lan2= 4 ,date= 2018/04/14 17:03:17 +0900 ,smac= bc:5c:4c:5d:1a:bf ,dmac= b8:27:eb:3a:6b:fa ,sip= 192.168.13.160 ,dip= 192.168.2.102 ,sha1payload= 9dac7a7beb944a7193847a3d0fbcc370d13a5838 ,payload = broadcast id

object page の中には LAN-3 と LAN-4 で p2p 通信が行われた情報は見当たらなかった．

4. 4　関連研究

4.1 AAFID

Autonomous Agents for Intrusion Detection (AAFID)[11]は我々の良性 botnet と同様に分散 IDS の agent の集合である．このシステムは，agents と transceivers と monitors で構成されている．AAIF の agent と我々の agent bot は，どちらともコマンドによって制御され，通信データを収集する部分で類似している．Agent AAFID の agent は client host に install されているのに対して，我々の agent bot は LAN とそのルータ又は NAT ルータの間に設置される．我々の良性 botnet の管理者は agent bot を client host のそれぞれに install する必要はない．AAFID の monitor と transceiver は，双方とも agent などからデータを集めて，それを解析する部分で類似している．AAFID の monitor は wiki ページの script で制御されないのに対して，我々の agent bot や analyzing bot は wiki ページの script で制御される．Agent 間の通信方式については，AAFID については定義されていないのに対して，我々の botnet は wiki API を使っている．

4.2 Man in the Middle Attack

我々の agent bot は一種の man in the middle attack[3] を行っていると解釈することもできる．Agent bot によって，sub-LAN 内の多くの通信を制御可能である．我々は Agent bot が dark side に行かないように注意する必要がある．

4.3 KASEYA and UNIFAS

Kaseya のパソコン管理システム[8]や Furuno Systems の無線 LAN アクセスポイント管理システム(UNIFAS)[12]は beneficial botnet と同様に，定期的にエージェントプログラムが Web サーバにアクセスし，そこに書かれた指示をエージェントが実行し，結果を Web サーバ側に戻すことを行っている．エージェントプログラムと Web サーバは NAT を超えて相互に通信できる．しかしながら，これらのシステムはセキュリティ強化を目的としたものではない．また，これらのシステムに特化した Web サーバを必要とする．

5 おわりに

現在試作を行っている悪性 botnet 包囲網(beneficial botnet)によって，マルウェアの P2P 通信を検出できる可能性があることを示した。我々の beneficial botnet は script を格納するための wiki ページとその script の interpreter から構成されている．評価実験を行う為，悪性 botnet の通信をまねる偽 Gameover ZeuS も作成した．

現時点で LAN-WAN 間通信が非常に遅いので実用的に使う為にはこの問題を改善しなければならない．セキュリティの強化とその検証も必要である．この他，利用しやすくするために，デバッグの方法や環境についても改善する必要がある．

謝辞

本研究の一部は JSPS 科研費 16K00197 の助成を受けて実施しました．良性 botnet およびその開発時に利用した PukiWiki, Java, Pcap4J, Eclipse, Eclipse Egit, M2Eclipse, Apache, Apache http client, twitter4j, Raspberry Pi, Raspbian の開発者，実験の実施を手伝ってくれた学生諸君に感謝します．

参考文献

[1] Andriesse, D. and Bos, H. “An Analysis of the Zeus Peer-to-Peer Protocol”, Technical Report IR-CS-74, rev. April 10, 2014.
[2] Conficker, https://en.wikipedia.org/wiki/Conficker
[3] Conti, M., Dragoni, N. and Lesyk, V. 2016 “A Survey of Man In The Middle Attacks”, 2016, IEEE Communications Surveys & Tutorials, Vol. 18, Issue 3, IEEE, 2027-2051. DOI=10.1109/COMST.2016.2548426
[4] FBI. 2014. GameOver Zeus Botnet Disrupted, https://www.fbi.gov/news/stories/gameover-zeus-botnet-disrupted, June 2, 2014.
[5] Ihaka, R., and R. Gentleman. “R: a language for data analysis and grapics”. 1996, J. Comp. Graph. Stat. 5:299-314. Available via http://www.R-project.org.
[6] JPCERT コーディネーションセンター, “P2P 型ボット分析レポート”, 2007, https://www.jpcert.or.jp/research/2007/P2P_bot_analysis_report.pdf
[7] “Microsoft Office 製品情報“. https://office.microsoft.com/ja-jp/products, (参照 2016-02-20).
[8] KASEYA, http://www.kaseya.com/
[9] Pcap4J, https://www.pcap4j.org
[10] Puri, R. “Bots & Botnet: An Overview," 2003, SANS InfoSec Reading Room, http://www.sans.org/rr/whitepapers/malicious/
[11] Spafford, E. H. and Zamboni, D., “Intrusion detection using autonomous agents”, 2000, Elsevier, Computer Networks vol.34, pp.547-570.
[12] UNIFAS, http://www.furunosystems.co.jp/product/detail/unifas.html
[13]Yamanoue, T., Oda, K., Shimozono. K., “Capturing Malicious Bots using a Beneficial Bot and Wiki”, 2012, In Proceedings of the 40th annual ACM SIGUCCS conference on User services (Memphis, Tennessee, USA. 15-19 Oct. 2012). ACM, New York, NY, 91-96. DOI=https://doi.org/10.1145/2382456.2382477
[14] Takashi Yamanoue, Kentaro Oda, Koichi Shimozono. “A Malicious Bot Capturing System using a Beneficial Bot and Wiki,” 2013, Journal of Information Processing (JIP), vol.21, No.2, pp.237-245.
[15] Yamanoue, T., Oda, K., Shimozono. K. 2013. “An Inter-Wiki Page Data Processor for a M2M System, “ 2013, In Proceedings of the 4th International Conference on E-Service and Knowledge Management (ESKM 2013), Advanced Applied Informatics (IIAIAAI), 2013 IIAI International Conference on.(Matsue, Shimane, Japan, 31 Aug- 4 Sep. 2013) IEEE, Los Alamitos, CA. 45-50. DOI= https://doi.org/10.1109/IIAI-AAI.2013.48
[16] Yamanoue, T., Oda, K., Shimozono. K., “Experimental Implementation of a M2M System Controlled by a Wiki Network,” 2014, In Applied Computing and Information Technology,Studies in Computational Intelligence, Springer, Vol.553, 121-136.
[17] Yamanoue, T., “Monitoring Servers, With a Little Help from my Bots, “ 2017, In Proceedings of the 45th annual ACM SIGUCCS conference on User services (Seattle, Washington, USA. 01-04 Oct. 2017). ACM, New York, NY, 173-180. DOI=https://doi.org/10.1145/3123458.3123461