ボットネットの特性解明に向けてのIDSルール作成とその検討

渡邊 昌幸 †　　安村 通晃 †　　小池 英樹 ‡

† 慶應義塾大学政策・メディア研究科
252-8520 神奈川県藤沢市遠藤 5322
{masayuki, yasumura}@sfc.keio.ac.jp

‡ 電気通信大学大学院情報システム学研究科
182-8585 東京都調布市調布ヶ丘 1-5-1
koike@acm.org

あらまし 近年，ボットネットはさらなる広がりを見せている．現状では，インターネットで観測されるボットネットは IRC(Internet Relay Chat) を利用した C&C(Command and Control) サーバーによって制御されるものがほとんどであり，P2P を利用したボットネットは流行の兆しがない． [1].

本研究では，ハニーポットを用いた定点観測によってボットネットの C&C サーバーに関する情報を抽出し，ネットワーク型侵入検知システム (NIDS) である Snort のルールファイルを作成した．また，ルールの有効性について検証し，ボットネットの特性解明のための考察を行った．

A Study of Characteristics of Botnet through Building IDS Rules

Masayuki Watanabe†　　Michiaki Yasumura †　　Hideki Koike ‡

†Graduate School of Media and Governance, Keio University
5322 Endo Fujisawa Kanagawa 252-8520, JAPAN
{masayuki, yasumura}@sfc.keio.ac.jp
‡Graduate School of Information System The University of Electro-Communications
1-5-1, Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
koike@acm.org

Abstract Nowadays, Botnets become prevailed, and most of the Botnets use IRC (Internet Relay Chat) channels as C&C server to connect malicious hosts to command and control. We observed fixed point Honeypot and builded some Snort rule files by extracting C&C information. We inspected effectivity of the rules and consider about characteristic of Botnet.

1　はじめに

インターネットを通じたマルウェア感染によるさまざまなセキュリティの問題は，ますます深刻化している．P2P を通じたマルウェア感染による情報流出が社会的問題化しているほか，近年，特にボットネットによるスパムメールの送信や DDoS 攻撃の問題が顕在化している．ボットの特徴として，ボットマスターと呼ばれる命令者が，C&C サーバーを利用して感染ホストに対して指令，制御を行うことが挙げられる．また，次々と亜種を生み出すことで既存のウイルス対策ソフトウェアによる検知を困難にしているほか，ダイナミック DNS の利用により C&C サーバーの冗長性を確保している．

P2P ボットの出現が予見されているが，P2P 機能を主通信機能として用いることは既存のプロトコルとの判別が容易であることや，P2P ネットワークの構築には一定数以上のノードが必要とされることから，依然として IRC を利用したボットネットが主流である．

本研究は，ハニーポットで収集したマルウェアの検体を仮想 OS 上で実行することによって C&C サーバーの情報を抽出し，ネットワーク型侵入検知システム (NIDS) である Snort のルールを作成すること

を通じてボットネットの特性を解明することを目的とする．

本研究の特徴は，「同じドメイン名の C&C に接続を行うマルウェアは，同一の作者またはグループによるボットである」という仮定のもと，ハニーポットで捕獲したマルウェアの接続先ドメインに着目し，Snort のルール作成及び検知実験を行っている点である．また，Snort のルールによるボットネット検知の可能性を探るため，接続先ドメインに着目し，タイムラインに沿ったボットネットの検知率シミュレーションを行なった．

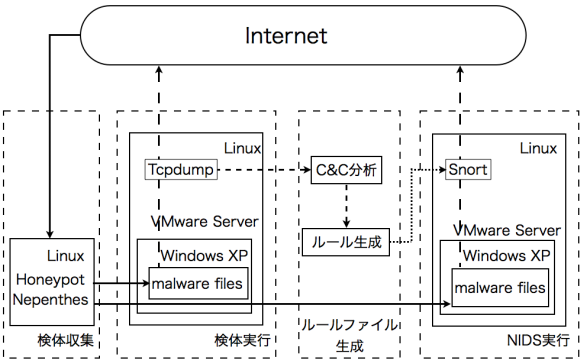

図 1: システム構成図

2　先行研究

須藤ら [2] はボットネットの捕捉，spam，DDoS 等の補足に有効な，ハニーポットと仮想インターネットを組み合わせたシステムを提案し，評価した．堀合ら [3] はボットネットの挙動解析支援システムを構築するとともにログ情報を視覚化し，ボットネットの特徴的な挙動が把握できることを示した．また，C&C サーバーとの通信に使用されるポートを制御することでボットネットの影響を緩和できる可能性を示した．

本研究では先行研究をふまえ，ボットネットの解析データを，侵入検知システムとしてネットワーク管理に広く普及している Snort に応用した．また，C&C サーバーのドメインとポート番号の組み合わせに着目し，Snort を用いたボットネット検知の可能性と，ボットネットの特性解明に向けて考察を行なった．

3　システム構成

3.1　システム概要

今回実験を行ったシステムの構成図を図 1 に示す．システムは検体収集部，検体実行部，ルールファイル作成部，NIDS 実行部で構成される．

3.2　検体収集部

ハニーポットの一種である Nepenthes [4] を用いて検体の収集を行なう．Nepenthes は Windows の脆弱性を模擬し，脆弱性を狙って攻撃してきたマルウェアを捕獲することができる．また，Nepenthes は捕獲した検体のハッシュ値を生成し，ハッシュ値をファイル名にして検体を保存する．

3.3　検体実行部

検体収集部で得た検体を仮想マシン VMware [5] 上の Windows で動作させる．検体とインターネット間の通信をすべてキャプチャし，ファイルをルールファイル作成部へ渡す．

3.4　ルールファイル作成部

ルールファイル作成部は C&C 分析部とルール作成部から成る．

C&C 分析部では，まず検体とインターネット間の通信が IRC プロトコルを使用したものかどうかを判断する．IRC プロトコルを使用したものであると判断した場合は，当該検体の接続先ドメイン名，ポート番号，接続に使用した IRC のニックネーム，ユーザ名を記録する．

今回，検体がインターネットとの通信に IRC プロトコルを使用しているかどうかの判断基準は，パケット中に”NICK”,”JOIN”,”USER”の文字列があることを基準とした．ボットネットに使用されている IRC サーバーは，IRC プロトコルが改変されている場合が多いが，これらの文字列については改変されずにそのまま使用されている場合がほとんどである．そのため，今回はパケット中の文字列のみで，検体が IRC サーバーと通信を行っているかどうか判断した．

ルール作成部では，C&C 分析部において IRC サーバーと通信を行なっていると判断された検体について，検体ごとに記録されたドメイン名とポート番号から Snort のルールファイルを作成する．

ルール作成に IP アドレスではなくドメイン名を使用する理由としてダイナミック DNS がボットネットに広く使用されており，C&C サーバーの IP アドレスが頻繁に更新されていることが挙げられる．本研究で得られたボットネットの C&C サーバーのドメイン名に対する IP アドレスの変化を観測したところ，国境を越えた IP アドレスの変化を見せる C&C サーバーも存在することから，IP アドレスベースでのルールファイル作成は効果的ではないと言える．

3.5 NIDS 実行部

NIDS 実行部では，ルールファイル作成部で作成されたルールファイルを適用した Snort を用い，仮想マシン上で動作させた検体の検知実験を行なう．

4 実験

図 2 に検体を収集した拠点及び期間を示す．本研究では，2007 年 1 月 18 日から 5 月 8 日までの 121 日間に拠点 A において収集した検体を用いて Snort のルールファイルを作成した．その後，拠点 A において 2007 年 5 月 9 日から 9 月 6 日までに収集した検体，および拠点 B において 2007 年 6 月 25 日から 9 月 6 日までに収集した 476 検体に対し，作成したルールファイルを適用した Snort で検知実験を行なった．

4.1 検体収集

2 拠点 A,B において検体収集を行なった．収集した検体について，表 1 に示す．

表 1: 収集した検体

期間番号	拠点名	収集期間	収集日数	検体数
1	A	2007.1.18 - 5.8	121	982
2	A	2007.5.9 - 9.6	121	868
3	B	2007.6.25 - 9.6	74	476

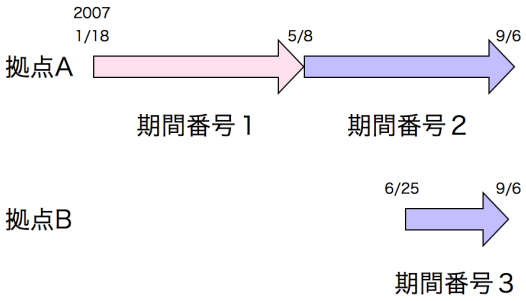

図 2: 検体収集拠点および期間

4.2 検体実行

ルールファイル作成のため，2007 年 1 月 18 日から 5 月 8 日にかけて収集した検体を動作させた．予備実験の結果などから，ボットネットを構成する検体は，ほとんどの場合動作開始から 1 分以内に C&C である IRC サーバーとの接続を確立することを確認していたため，検体の実行時間は 1 分とした．

また近年，VMware などの仮想マシン検知技術の発展により，正規の挙動を行わない検体の存在が報告されているが，本研究においては検体による仮想マシンの検知を考慮しないこととした．

4.3 ルールファイル作成

検体実行によって得られたファイルを C&C 分析部で分析することにより，62 箇所の C&C サーバーのドメイン名と，使用されているポート番号の組み合わせを得ることができた．この情報を元に作成した独自ルールの例を以下に示す．

alert tcp $HOME_NET any -> xx.enterhere.biz 10324 (msg:"This is own alert couse by A information."; sid:1100002;)

4.4 Snort による検知実験結果

作成した独自ルールを適用した Snort を使用し，仮想マシン上で動作させた検体の検知実験を行った結果を表 2 に示す．検体の実行時間はルール作成時と同じく 1 分とした．

表 2: Snort を用いた検知実験結果

拠点名	検体総数	検知数	検知率 (%)
A	868	191	22.0
B	476	65	13.7

実験に用いた拠点 A の検体は，5 月 9 日以降に収集した検体のうち，5 月 8 日以前に収集した検体と同じハッシュ値を持つものを除いた．また，今回作成した Snort の独自ルールでの検知のみを検知数としてカウントした．なお，作成した 62 の独自ルールのうち，19 のルールは C&C サーバーのドメイン名に対する DNS の応答がなくなったため，ルールとして適用することができなかった．これは，5 月 8 日の検体実行時には存在した C&C サーバーが，9 月 6 日の時点では何らかの理由で存在しなかったことによるものである．そのため，実際には 43 の独自ルールを適用した状態での実験となった．

5 考察

5.1 独自ルールの有効性

本実験は，拠点 A のある時点 (5 月 8 日) まで (期間番号 1) の情報を元に作成したルールが，

- ルール作成時以降 (5 月 9 日以降) の拠点 A(期間番号 2) への攻撃
- IP アドレス帯が異なる拠点 B(期間番号 3) への攻撃

に対して，どの程度有効であるかを検証したものである．

本実験では，ルール作成以前に収集した検体と同じハッシュ値を持つ検体を 5 月 9 日以降に観測した場合には除外している．そのため本実験の結果は，5 月 9 日以降に観測した新たなハッシュ値を持つ検体に対しての検知率である．また，Nepenthes で収集した検体は必ずしもボットネットを構成するものとは限らず，本ルールでは根本的に検知不可能な単純なワームなどの検体も含まれていることを考慮に入れる必要がある．

A 拠点では 43 の独自ルールで 191 の検体を検知することができるとの結果を得たが，これは 43 のボットネットが，本研究におけるルール作成以降 (5 月 9 日以降) に，191 個ものハッシュ値の異なる新たな検体を作成して感染活動を行なわせていたという事実を示していると考えることができる．

ボットネットが，シグネチャベースのアンチウイルスソフトの検知を逃れるためにボットプログラムを更新，または亜種を作成して拡散させることは既知の事実であった．本実験によって，C&C サーバーのドメイン名とポート番号の組み合わせを用いることで新たなボットの亜種に対して今回の独自ルールが有効であることが確認できた．

5.2 独自ルール作成方法の妥当性

今回のルール作成方法は，検体実行のパケット内に”NICK”や”JOIN”などの IRC 固有の文字列が存在した場合にその検体をボットネットを構成する検体と判断し，接続先のドメイン名とポート番号を用いてルールを作成するというものであった．

ボットネットに使用されている IRC サーバーは第 3 者の C&C サーバーへのアクセスを困難にするため，既存の IRC クライアントでの接続をできなくするなど，IRC のプロトコルが改変されている場合が多い．また，通常 IRC サーバーで使用できる LIST や NAMES などのコマンドが使用できなくなっているほか，IRC では通常 6667 番を使用しているポート番号についても不規則な番号を用いたり，80 番や 433 番を使用する場合がある．

Snort には chat.rules というルールセットがあり，パケット内に”NICK”や”JOIN”などの IRC 固有の文字列を発見した場合にアラートを返すルールが存在する．しかし，このルールは本来 IRC でのチャットを検知する目的のためか，ポート番号を本来 IRC が使用するポート番号に限定しているため，近年の IRC プロトコルを改変したボットネットの検知は困難である．

また仮に，ポート番号を限定せずにパケット内の IRC 固有の文字列を検知する Snort のルールを作成した場合，本研究と同じ環境であれば，今回の実験と同程度の検知率があるものと考えられる．しかし，実際にネットワーク管理の現場でそのルールを使用した場合，本来の IRC 通信をも検知してしまうため，誤検知が多すぎて実用に適するルールにはなり得ない．

5.3 ルールファイルの有効期限

今回の実験で，5 月 9 日に作成した 62 のルールのうち，19 のルールが 9 月 6 日の時点で無効になって

いた．ボットネットの C&C サーバーのドメイン名を指定してルールを作成する方式の場合，C&C サーバーが存在しない，または C&C サーバーに対して既に IP アドレスが割り当てられてないことを考慮する必要がある．このような事象が発生する原因としてはボットネットの管理者 (ボットマスター) が，ダイナミック DNS を利用して，当該ドメイン名に対する IP アドレスの割り当てを意図的に解除した場合や，ISP などの意図で，C&C サーバーのドメイン名が無効になった場合などが考えられる．後者の場合は，当該ドメイン名を Snort のルールから削除して Snort の運用を継続することが問題ないものと考えられるが，前者の場合は，ボットマスターがダイナミック DNS を利用して，当該ドメインに対して再度 IP アドレスを割り当てる可能性がある．

ルールファイルを常に最適な状態に保つためには，検体実行によって得られた C&C サーバーのドメイン名に対する IP アドレスの変化を継続的に監視していく必要がある．

5.4 最適なルールファイル作成

Snort は起動時に DNS クエリーによってルール内ののドメインに対する IP アドレスを得，流れているパケット内の IP アドレスと参照することでアラートなどのアクションを起こしている．起動時の DNS クエリーに対し，IP アドレスを返さないドメインが存在した場合，Snort はエラーを出力して起動を停止する．そのため，今回のようにルールファイルにドメイン名を記述する場合，そのドメイン名が Snort 起動時に有効でなければならない．

また，ボットネットはダイナミック DNS を用いることでドメイン名に対する IP アドレスを頻繁に更新している．そのため Snort 起動後，ボットネットの C&C サーバーの IP アドレスが変化すれば，ルールファイルが有効でなくなり，そのボットネットを検知することはできなくなる．

これらの問題を回避するためには C&C サーバーが有効な IP アドレスを返すかどうか定期的に観測を行うことで，常に新鮮なルールファイルを作成し，適用することが必要である．今回の実験から，最適なルールファイルの作成方法について，図 3 に示す．

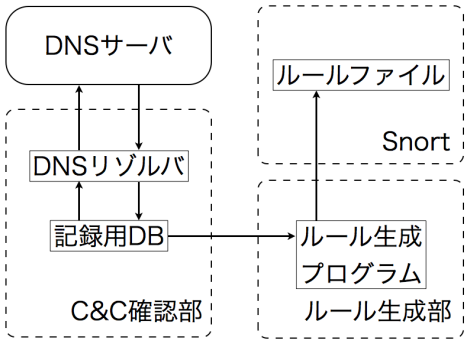

図 3: ルールファイル作成法

5.5 リアルタイムルール作成のシミュレーション

仮想 OS 上で検体を実行して得た C&C に関する情報に基づき，以下の想定のもとで Snort のルールを作成した場合の Snort の検知率の変化をシミュレーションにより検証した．

- ハニーポットで捕獲したマルウェアの情報を瞬時に Snort のルールに反映することができる
- C&C サーバーはボットと常に通信可能な状態にある

シミュレーションではハニーポットのログを時系列順にたどる．事前の分析結果から，C&C サーバのドメイン名とポート番号を得ることができる検体からの攻撃であった場合，ボットとして検知可能であると判断する．

拠点 A，B のそれぞれで Snort による C&C 検知率の変化を計算した．拠点 A，B における C&C 検知率の変化をそれぞれ図 4, 図 5 に示す．

検知率は拠点 A，B ともに初日に高い値を示し，その後収束するという傾向が得られた．常に初日に高い検知率が得られるとは限らないが，いずれの拠点の場合も開始から約 1ヶ月間で検知率がある程度の値に収束するという結果が得られた．

期間中のハニーポットに対する感染活動全体の攻撃数と検知可能検体による攻撃数を表 3 に示す．

表 3: 全攻撃数と検知可能な検体による攻撃数

拠点名	全攻撃数	検知可能検体攻撃数	検知率 (%)
A	38469	11383	29.6
B	18513	3050	16.5

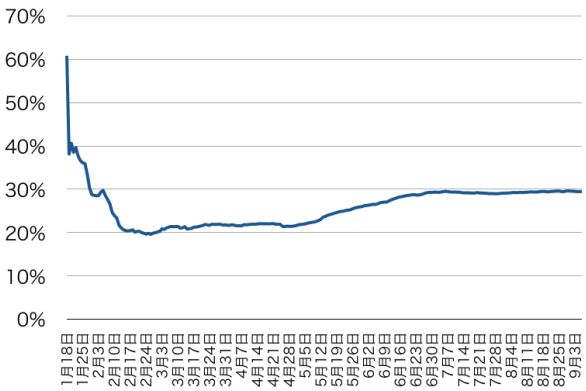

図 4: 拠点 A における検知率の変化

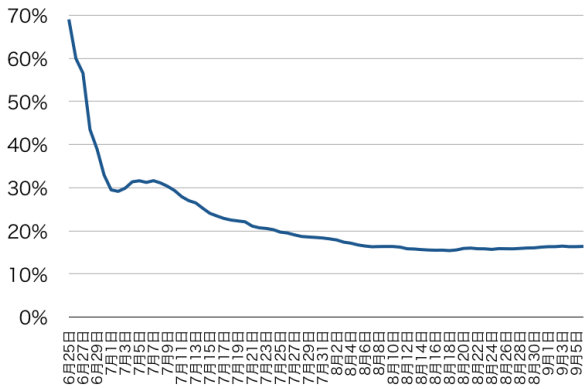

図 5: 拠点 B における検知率の変化

また，期間中に捕獲した全検体数と，本方式で検知可能な検体数について以下に示す．

表 4: 捕獲した全検体と検知可能な検体

拠点名	全検体数	検知可能検体数	検知率 (%)
A	1857	751	40.4
B	476	116	24.4

このシミュレーションの結果から，リアルタイムで Snort のルールを作成し適用した場合, 検知可能な検体の割合は拠点 A において 40.4%，拠点 B において 24.4%となっている．表 2 に示した，実験結果に比べて検知率が向上しており，リアルタイムでルールを作成することの有効性について確認することができた．ただし，この検知率上昇はハニーポットで捕獲したマルウェアの情報を瞬時にルールに反映できるということによって，表 2 の実験結果よりもルールの作成対象となる期間が長くなったことによるルール数の増加が主な原因と考えることもできる．本シミュレーションを実験によって検証する場合，C&C サーバーの消滅などの不確定要素や，検体実行から Snort のルール適用までのタイムラグを考慮に入れて実験を行なう必要がある．また，シミュレーションでは約 1ヶ月間検知率が安定しなかったことから，Nepenthes 等のハニーポットで検体を収集して Snort のルールに生かす場合，最低でも 1ヶ月の検体収集期間が必要ではないかと考えることができる．

6 おわりに

本研究では，Snort のルール作成を通じたボットネットの特性解明のため，Snort の独自ルール作成と独自ルールを用いたボットネット検知実験，およびリアルタイムでルールを作成した場合の Snort によるボットネット検知率のシミュレーションを行った．

本研究によって，IRC を前提としたボットネット C&C サーバーを，ドメイン名とポート番号を用いた Snort のルールによって検知することの有効性を示した．また，リアルタイムで Snort のルールを作成した場合のシミュレーションを行い，リアルタイムでルールを作成することの有効性を示すとともに，ボットの検体を収集して Snort のルールを作成する場合に必要な検体収集期間についての知見を得ることができた．

今後は，C&C サーバーの移動や消滅などの，ボットネットの動的側面を考慮しつつ，今回行なったシミュレーション結果を実験によって確認し，さらなる特性解明を進めていく．

参考文献

[1] P2P 型ボット分析レポート: JPCERT CC (http://www.jpcert.or.jp).

[2] 須藤年章，富士原圭: 仮想インターネットを用いたボットネット挙動解析システムの評価, CSS2006.

[3] 堀合啓一，大橋洋一，朝長秀誠，田中英彦: 定点観測によるボットネットの挙動観測とログ情報の視覚化, SCIS2007.

[4] Nepepnthes http://nepenthes.mwcollect.org

[5] VMware http://www.vmware.com