通信の状態遷移に着目したボット活動の調査

水谷 正慶 †　　武田 圭史 †　　村井 純 ††

† 慶應義塾大学大学院 政策・メディア研究科
†† 慶應義塾大学 環境情報学部
252-8520 神奈川県藤沢市遠藤 5322

{mizutani,keiji,jun}@sfc.wide.ad.jp

あらまし　本稿は複数のボットに共通する通信の状態遷移モデルに基づいてボットの活動を調査した，研究用データセット CCC DATAset 2008 の攻撃通信データにおけるボット活動に関する調査について述べる．ボットは従来のコンピュータウィルスと比較して短期間で亜種が発生し，各ボットの特徴が変化してしまう．しかし，多くのボットには共通した目的の動作の流れが存在するため，これを状態遷移として捉えられれば未知のボットの検知に利用できると考えられる．本稿ではボット感染直後の通信の状態遷移モデルを作成し，モデルに基づいて CCC DATAset 2008 を解析した．そして，ボットが通信などで失敗しない限り，モデルに沿って動作することを確認した．

A Study of Bot Activities Focused on State Transition of Communications

Masayoshi Mizutani†　　Keiji Takeda †　　Jun Murai††

†Keio University - Graduate School of Media and Governance
††Keio University - Faculty of Environment and Information Studies
5322, Endo, Fujisawa-shi, Kanagawa 252-8520, Japan

Abstract　This paper describes a study of controlled malicious softwares (Bots) activities included in CCC DATAset 2008 based on our state transition model of common bots. Bot variants are generated for a shorter period than traditional computer viruses and bot's features are frequently changed. We focused common sequence of bot's actions and built a state transition model soon after bot infections to detect new bot variants. In this paper, we analyzed CCC DATAset 2008 based on our model and confirmed that bots activities are accooding to our model if bot communications are not failure.

1　はじめに

ネットワーク上で発生する情報資産の侵害を検出するために，多くのセキュリティ管理者は侵入検知システム (IDS) によってネットワークセキュリティに関する事象を検知している．既存の IDS は第三者から命令をうけて動作する悪意あるプログラム (ボット) を検知する際，固有の通信パターンを定義したシグネチャを用いている．しかし，ボットは従来のコンピュータウィルスなどと比べると非常に短期間で亜種が発生する傾向がある [1]．そのため新しい亜種が発生した場合，ボットの検知に利用するシグネチャが更新されるまで検知ができないという問題がある．

本稿ではボットの通信の類似性に着目し，複数のボットに共通する活動の存在を示す．ボットは短期間で亜種が発生するが，基本的な動作の流れは大きく変更されない傾向が見られる．個々の動作は一般的な通信プロトコルやボット独自のプロトコルを利用するなど，ボット毎の固有のパターンや組み合わせが存在し，亜種の発生によって多様に変化している．

しかし複数の動作を，一連の目的にそった状態遷移として捉えた場合，多くのボットに共通する動作の流れがあると考えられる．ボットは感染後に Command & Control(C&C) サーバから命令を受信し，ボットネット管理者が自由に感染ホストを操作できるのが特徴である．これを実現するために実行ファイルのダウンロードや C&C サーバへの接続が必要であり，これらの動作の流れの共通項を調査した．

複数のボットの通信に類似する状態遷移が明らかになることで，通信の状態遷移や相関関係を利用したネットワークセキュリティ監視手法 [2] が利用できる．既存の IDS では厳密にボットの通信内容をシグネチャで定義しなければ誤検知を多発してしまうため，多くのボットに対応するルールを作成するのは難しかった．本稿でボットに共通する検知ルールを作成できれば，少ないルールで多くのボットに対応できるだけではなく，新たに発生した亜種に対しても有効なルールになると期待される．

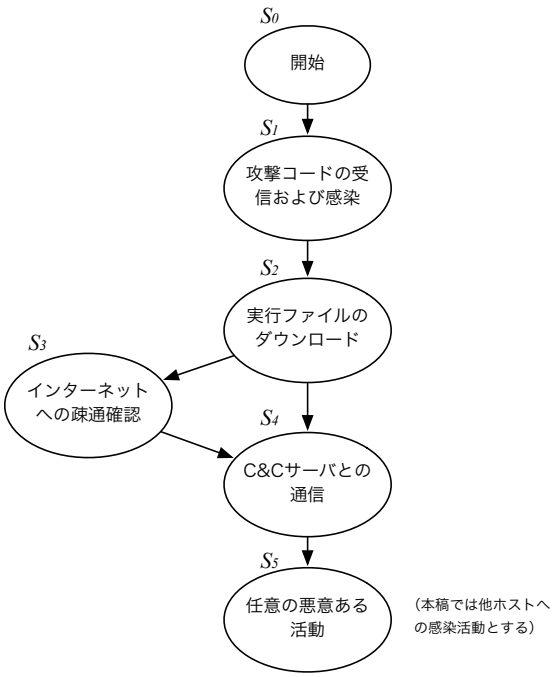

図 1: 感染直後のボット活動の状態遷移モデル

2　ボットの活動モデル

本稿ではまずボットの感染直後の活動を表す状態遷移のモデルを示し，研究用データセット CCC DATAset 2008 の攻撃通信データ (以下，CCC DATAset) を用いてモデルの正当性を検証する．CCC DATAset の通信データにはセキュリティ対策が十分ではないホストをインターネットに接続し，ボットの感染と感染後の活動が含まれる．感染後の活動記録は限られた時間しか記録していないため，感染の成功と感染直後のモデルに限定して議論する．

本稿では実行ファイルを利用し C&C サーバから指示をうける悪意あるプログラムをボットと定義する．個々のボットが複雑な動作をするため，ホストの脆弱性を利用して不正利用するために送信される攻撃コードとは別に専用の実行ファイルが別途ダウンロードされる場合が多い．そのため，感染活動だけを実行する実行ファイルを持たない Slammer Worm[3] などは含まない．また，自律的に他ホストへの感染活動を繰り返すプログラムもボットに含めない．

上記の条件に基づき作成したボットの活動モデルを図 1 に示す．図 1 ではボット感染後の通信に関する状態遷移を示している．状態は S_0) 開始状態，S_1) 攻撃コードの受信と感染，S_2) 実行ファイルのダウンロード，S_3) インターネットへの疎通確認，S_4)C&C サーバとの通信，S_5) 任意の悪意ある活動の 5 種類とした．各状態におけるボットの具体的な通信内容は，ボットの種類に応じて異なるものとする．例えば，実行ファイルのダウンロードでは HTTP[4]，FTP[5]，TFTP[6]，あるいはボット独自のファイル転送用プロトコルが挙げられるが，各通信がボットとして活動するための実行ファイルをダウンロードするという共通の目的であるとする．

今回使用する通信データは，感染元のホストが監視対象のホストと通信することでボットに感染する．そのためボットに感染した場合は，感染後にはなんらかの方法で実行ファイルをダウンロードする．実行ファイルをダウンロードした後は C&C サーバと通信する．ただし，近年はボット研究のために情報セキュリティ関係の研究者が閉鎖されているネットワーク上でボットを実行する場合がある．C&C サーバとの通信前にインターネット上の著名な Web サービスなどへ接続することで，疎通を確認するボットが存在する．また，S_5 はスパムメールやトラックバックスパムの送信，感染ホストに関する情報の報告，Web サービスのアカウント不正取得などが挙げられるが，本稿では C&C サーバとの通信直後に発生しやすい活動として，他ホストへの感染活動に着目した．

3 CCC DATAset を用いたボット活動の分析

3.1 データセットの分割と解析

第 2 節で作成したモデルを検証するために，ボットによる通信データを調査した．調査に利用した CCC DATAset の通信データには Windows 2000 と Windows XP のホストをそれぞれ 1 台ずつインターネットに接続し，攻撃活動に関する通信が含まれる．両ホストは定期的にクリーンな状態にリセットされている．今回のモデルは正常状態からボットへの感染，ボットとしての活動開始までの遷移に着目するため，OS が初期状態に復元された時刻を正確に把握しなければならない．一方の対象ホスト (以降、H_1) は Windows XP であり，起動時に time.windows.com. への問い合わせ，および NTP による時間同期を実行する。そのため，通信データから H_1 が送信元もしくは宛先ホストとなっているパケットを抽出し，前回のリセット予想時刻から一定時間以上経過した後に発生する time.windows.com を問い合わせる DNS パケットと，NTP のパケットによって通信データを分割した．これによって 144 個の通信データセットが生成された，

CCC DATAset2008 を復元時刻によって分割した全てのデータセットについて，状態遷移を調査した．調査した結果を表 1 に示す．複数のボットに感染したと見られるデータセットもあるが，表の状態遷移では最初にボットに感染した際の状態遷移を示している．それぞれのデータセット群を D_N として分類した．ただし，D_0 では注目すべき通信が発生せず，D_1 は攻撃が成功しなかった可能性が高いため除外して議論する．また，状態遷移のモデルに従わなかったものを D_e とした．これは C&C サーバとの通信をせずに自律的に感染活動を実施しており，本稿におけるボットの定義とは異なる．データセットの状態遷移では，各データセットは図 1 で示したどの状態を遷移したのかを示している．結果の欄ではボットが最後の状態から最終的にどのような活動に至ったかを示している．最後の状態以降にボットとしての活動と思われる通信が観測されなかった場合が**終了**，観測されたデータセットが**継続**，終了に加えて最後の状態での活動に不備があったと見られる痕跡が観測されたデータセットが**失敗**としている．

表 1: 分割したデータセットの分類

	データセットの状態遷移	結果	件数
D_0	S_0	終了	11
D_1	$S_0 \rightarrow S_1$	終了	9
D_2	$S_0 \rightarrow S_1 \rightarrow S_2$	失敗	26
D_3	$S_0 \rightarrow S_1 \rightarrow S_2$	終了	5
D_4	$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_4$	失敗	1
D_5	$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_4$	継続	13
D_6	$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_4 \rightarrow S_5$	継続	51
D_7	$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5$	継続	27
D_e	モデルから外れたデータセット	----	1
		合計	144

3.2 状態遷移に基づいた分析

表 2 では各状態が遷移する際の時間間隔として，各状態遷移のサンプル数，平均時間，標準偏差，サンプル中の最小時間，サンプル中の最大時間を示している．今回の分析では状態遷移の時間を，遷移前の状態における最後の動作の時刻と遷移後の状態における最初の動作の時刻の差としている．これは各状態では複数回類似した動作が繰り返される場合があり，通信データだけではどの動作が次の状態遷移へのきっかけになったかを判別するのが難しいためである．例えば，実行ファイルがダウンロードされる前に攻撃コードの受信する回数は，各データセットにおいて平均 3.2 回発生しており，成功した攻撃コードを特定するのは難しい．そのため，このような状態遷移を分析する場合には今後も検討が必要となる点に留意するべきである．

表では $S_1 \rightarrow S_2$，$S_2 \rightarrow S_3$，$S_2 \rightarrow S_4$，$S_3 \rightarrow S_4$ の状態遷移はそれぞれ平均時間が 15 秒未満となっており，最大時間もそれぞれ 60 秒程度である．これは，ボットがあらかじめ定められたアルゴリズムに従って単独で動作しているため，次の状態に遷移するための待ち時間を設ける必要がないためだと考えられる．各遷移において遷移時間が比較的長いサンプルがいくつか存在した．これらはホスト内部における情報の検索にかかる時間か，もしくは通信の間隔を長くしセキュリティ管理者に発見されにくくしていると予想される．

一方 $S_4 \rightarrow S_5$ は平均遷移時間が 137.091 秒，最大遷移時間が 489.683 秒であり，他の状態遷移と比較して間隔が長い．これは単独で実行しているプログラムに従った動作ではなく，IRC プロトコルによっ

表 2: 各状態開始から遷移までの所要時間

状態 遷移	数	平均遷 移時間	標準 偏差	最小遷 移時間	最大遷 移時間
$S_1 \rightarrow S_2$	96	5.116	11.401	0.041	66.416
$S_2 \rightarrow S_3$	27	2.688	9.765	0.219	51.451
$S_2 \rightarrow S_4$	63	4.618	11.050	0.282	66.416
$S_3 \rightarrow S_4$	27	11.334	16.190	0.006	59.280
$S_4 \rightarrow S_5$	78	137.091	147.392	1.096	489.683

(遷移時間の単位は秒，各計算結果の小数点第 4 位以下は切り捨て)

表 3: S_2 における通信内容の内訳

ダウンロード方法	結果	件数
独自プロトコルによる攻撃元ホスト からの実行ファイルのダウンロード	成功	83
	失敗	0
独自プロトコルによる攻撃元ホスト からの実行ファイルのアップロード	成功	7
	失敗	5
DNS による名前解決および HTTP による実行ファイルのダウンロード	成功	7
	失敗	0
バックドアによって攻撃元ホストが 指示した実行ファイルのダウンロード	成功	0
	失敗	20
TFTP による実行ファイルの ダウンロード	成功	1
	失敗	1

て C&C サーバから非同期的に送信されるコマンドに従っているためである．IRC による他ホストへの感染活動の実行命令例は [7] などに示されている．今回のデータセットでは活動時間が限られていたため，長期間の観測では遷移までの時間が長くなる可能性がある．

3.3 各状態における観測事象

第 2 節で述べたとおり，$S_{1,2,3,4,5}$ の各状態は具体的な通信内容を限定しない．そのため，CCC DATAset において観測された具体的な通信内容について述べる．

まず，S_1 では Windows OS で標準的に備わっている NetBIOS や DCERPC の脆弱性を利用した攻撃コードが送信され，感染している．ただし，CCC DATAset の通信データではホストの状態変化を知るためのデータがないため，どの攻撃によって感染したかを特定するのは難しい．そのため，S_2 へ遷移する直前の攻撃を S_1 の成功と見なした．

S_2 の通信内容について解析した結果を表 3 に示す．実行ファイルのダウンロードを発見するためには Windows で利用される実行ファイルのフォーマット [8] のパターンを検知するルールを作成し，検査した．このルールによって TCP および UDP による通信のペイロードを全て検査したため，暗号化されていない通信中の実行ファイルを検知できる．また，実行ファイルが検知されなかったデータセットについても目視で確認したが，暗号化などによって検知を回避した形跡は発見されなかった．独自プロトコルによるダウンロードおよびアップロードは全て同一のプロトコルであると見られる．これは第 3.4 節で詳しく述べる．バックドアによる実行ファイルのダウンロードでは，Windows のコマンドプロンプトを利用して FTP コマンドを発行しているが，実行ファイルのダウンロードは確認されなかった．TFTP による実行ファイルのダウンロード失敗では，サーバ側のホストが応答せず通信が終了していた．

S_3 では著名な Web サイトやメールサーバに接続を試み，インターネットへの到達性を確認していると予想される．観測された確認方法は主に 2 種類ある．1 つは著名なメールサービスの MX レコードを問い合わせた後，DNS の応答に含まれるメールサーバに対して SMTP[9] の標準ポート番号である 25 番へ接続を試みている．ただし，CCC DATAset の環境では内部から外部への TCP ポート 25 番がフィルタリングされていたと見られ，全てのメールサーバから応答が観測されなかった．もう 1 つの確認方法は Web サイトから特定のファイルをダウンロードする手法である．HTTP によるインターネットへの到達性確認には AOL のインスタントメッセンジャー [10] や ICQ[11] のインストールファイルが利用されていた．筆者が独自に観測したボットの活動では，ポート 25 番への到達性がある場合に迷惑メールの配信を開始したものがあるため，到達性確認後の活動にも影響していると推測される．

S_4 における C&C サーバとの通信は全て IRC プロトコルを利用していた．データセット $D_{4,5,6,7}$ において IRC 接続は合計 216 件発生していたが，TCP のサーバ側ポート番号は IANA で定められている標準のポート番号 [12] を利用していたのは 29 件のみであった．D_4 での接続失敗は DNS によって問い合わせた IRC サーバのアドレスから応答が無かったためである．ただし，別のデータセットでは同一のサーバとポート番号に対して IRC の接続が確立していたことから，サーバ側ホストの一時的な問題に

よって接続が失敗したと見られる．

S_5 における他ホストへの感染活動は S_1 での攻撃と同様に，Windows の NetBIOS や DCERPC の脆弱性を利用するためポート開放の確認と攻撃となっている．最初に観測されたスキャンの宛先ポート番号は TCP の 135 番が 71 件，139 番が 2 件，445 番が 6 件であった．

3.4 ボット独自の通信方法に関する分析

ボットは RFC などで定義されている一般的なプロトコル以外に独自のプロトコルを用いて相互に通信しており，CCC DATAset2008 においても独自のファイル転送プロトコルが確認された．感染直後に当該プロトコルを利用して実行ファイルをダウンロードしたのは 90 件だが，C&C サーバとの通信開始後に実行ファイルを更新したと見られるものを含めると，データセット中に同様のプロトコルが 351 件確認できる．データセット中で HTTP によるボットの実行ファイルダウンロードは 337 件であったため，当該プロトコルは CCC DATAset の通信データで感染した種類のボットにおいて，HTTP より多用されている．

プロトコルの概要を図 2 に示す．このプロトコルは 3 種類の通信で構成されており，それぞれ Hello，Ack，Data と呼ぶ．通信は TCP を用いて行われるが，サーバとクライアントでの区別ではなくファイル受信ホストとファイル送信ホストとして区別されている．TCP の SYN パケットをどちらが送信したかに関わらず，TCP セッションの確立後はファイルを送信する側が 380 バイトの固定長である Hello メッセージを送信する．一方，ファイルを受信する側のホストは 4 バイトの固定長である Ack メッセージを送信する．この 2 つのメッセージ送信が終了した後，送信側のホストが実行ファイルの送信を開始する．実行ファイルは任意の長さであり，実行ファイルの送信後に通信が終了する．

図 3 に Hello，Ack，Data の通信パケットの内容をそれぞれ示す．Hello，Ack はそれぞれ 1 パケットで送信されており，共に固定長である．Hello は先頭から 168 バイトと末尾 200 バイトは全ての通信において同じデータとなっており，この間の 12 バイトがファイル名の指定と考えられる．ファイル名は 12 バイト以内で拡張子を含み，ファイル名が 12 バイト未満の場合はファイル名以降が 0 となる．Ack

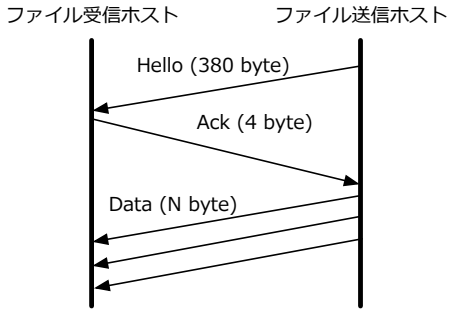

図 2: ボットの独自ファイル転送プロトコル概要

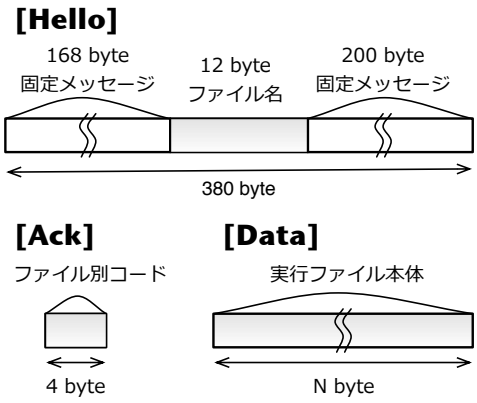

図 3: ファイル転送プロトコルのメッセージ

も 1 パケットで 4 バイトを送信しているが，こちらはデータが変化しており内容の推測は困難である．ただしデータセット解析の結果，Ack で送信されるデータとファイル送信ホスト，送信ホスト側の TCP ポート番号，送信されるファイルの内容が対応している事が明らかになっている．表 4 は Ack データと送信ホスト，送信元ポート，実行ファイルの種類の対応の 1 部を抜粋し示している．Ack データは送信ホスト，送信元ポート，実行ファイルの種類が同じ場合にのみ同一のデータが送信されている．ただし，これら 3 つのいずれかが異なる場合には，Ack データも異なった者が送信されている．これらの事実から Ack データは認証のキーに相当するものだと推測する．ただし，現在 Ack データの生成方法は明らかではない．Ack データの送信後は実行ファイルが送信される．

また，表 3 で述べているアップロードの失敗とは，TCP の接続は確立するが一方のホストが送った Hello，もしくは Ack のデータに対して他方のホストが応答しないという場合であった．

表 4: ファイル転送プロトコルにおける Ack データと他情報の対応

Ack データ (16 進数)	ファイル送信ホスト	送信元ポート	実行ファイルの種類
39 EC CE 28	aaa.6.253.95	4596	F_A
22 F4 2D F0	aaa.6.253.95	4596	F_B
22 F4 2D F0	aaa.6.253.95	4596	F_B
08 22 48 81	bbb.18.136.55	26777	F_C
B7 7D 5E 28	aaa.6.23.245	53177	F_D
F8 93 2E F8	bbb.18.226.12	16937	F_D

4 考察

調査した結果，ボットの活動が途中で不備を起こさない限りは図 1 にそって状態が遷移していることを確認した．$D_{5,6,7}$ のデータセットでは作成したモデルの状態遷移が確認された．攻撃コードの受信や実行ファイルの取得，インターネットへの接続性確認，C&C サーバへの接続はそれぞれ以前より確認されていたボットの活動だが，調査したボットではこれらが同一のモデルに従って動作する事が明らかになった．この事実は，状態遷移に従って活動するボットは亜種であっても検知できる可能性を示唆している．各状態の具体的な通信方法が異なっていても，これらを抽象化できる表現力を持ったネットワーク監視手法であれば，大幅に活動内容を変更しない限り検知できる．そのためシグネチャを作成していないボットの検知も可能となり，ボット対策として意義があると考える．

一方で，途中で動作が失敗してしまうデータセットやモデルに沿わないデータセットも存在する．各状態において失敗した動作については第節で述べたが，これらは総じてボットネットが他者の資源を無断で利用している動的なシステムであることに起因する．多くの場合，ボットは発見されるとそのホストから除去されてしまうため，C&C サーバや実行ファイル配布サーバが頻繁に変化する．ボットネット全体にこうした変更の情報が伝播せず，動作の失敗が起こりやすいと考えられる．しかし，今回のデータセットでは活動時間が限られており，通信データはボットが感染した直後の通信に限られている．そのため，感染から長時間経過した場合に新しいモデルを作成できる傾向が発見できる可能性がある．また，感染直後ではなく感染後にネットワークの変更や再起動などが実施された場合の活動に対しても新たな動作が見られる可能性があり，より様々な状況での通信データ収集が必要になると考える．

5 今後の課題

今回の CCC DATAset は定点による観測であり，さらに OS のリセットを正確に判断できるサンプル数が限られていた．定点による観測では同系統のボットによる感染を繰り返す可能性があり，サンプルに偏りが生まれてしまう．そのため，複数地点における観測データを解析・比較する必要がある．

さらに今後は，Web ブラウザなどを経由して攻撃コードを受信させる受動感染型ボットの感染活動にも着目し，モデルを作成する必要がある．今回利用した通信データは，全て感染元のボットが能動的に攻撃コードを送信するタイプである．受動感染型のボットは能動感染型のボットと比較して感染時の動作だけではなく，感染後の活動にも異なる部分が多いと予想される．

参考文献

[1] Paul Bacher,Thorsten Holz,Markus Kotter,Georg Wicherski. Know your enemy: Tracking botnets, May 2005. http://www.honeynet.org/papers/bots/.

[2] 水谷正慶, 白畑真, 南政樹, 村井純. 複数セッションの相関関係を利用したセキュリティイベント検知手法の提案. *IPSJ CSEC* **コンピュータセキュリティシンポジウム** *2007* **論文集**, 2007.

[3] CERT Advisory CA-2003-04 :MS-SQL Server worm. *http://www.cert.org/advisories/CA-2003-04.html*, Jan 2003.

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. *RFC:2616*, Jun 1999. http://www.ietf.org/rfc/rfc2616.txt.

[5] J. Postel. J. Reynolds. RFC:959, Oct 1985. http://www.ietf.org/rfc/rfc959.txt.

[6] K. Sollins. The TFTP Protocol (Revision 2). RFC:1350, Jul 1992. http://www.ietf.org/rfc/rfc1350.txt.

[7] Evan Cooke, Michael Bailey, Farnam Jahanian, Richard Mortier. The Dark Oracle: Perspective-Aware Unused and Unreachable Address Discovery. *The 3rd ACM/USENIX Symposium on Networked Systems Design and Implementation*, May 2006.

[8] Eelco Visser. PC Executable Format. Program-Transformation.Org: The Program Transformation Wiki, Apr 2000. http://www.program-transformation.org/Transform/PcExeFormat.

[9] Jonathan B. Postel. Simple Mail Transfer Protocol. RFC:821, Aug 1982. http://www.ietf.org/rfc/rfc821.txt.

[10] Instant Messenger AIM. http://dashboard.aim.com/aim.

[11] ICQ.com. http://www.icq.com/.

[12] Internet Assigned Numbers Authority. Port numbers, 2007. http://www.iana.org/assignments/port-numbers.