マルウェアの通信履歴と定点観測の相関について

小堀 智弘† 菊池 浩明† 寺田 真敏††

† 東海大学電子情報学部情報メディア学科
259-1292 平塚市北金目 1117 nopay,kikn@cs.dm.u-tokai.ac.jp
†† 日立製作所 Hitachi Incident Response Team (HIRT)
212-8567 神奈川県川崎市幸区鹿島田 890　日立システムプラザ新川崎

あらまし 本研究の目的は，ボットに感染しているホストから攻撃されるポートスキャンのパケットがネットワーク上で分散観測している非対話センサーで得られるものと同一かどうかを明らかにすることである．我々は CCC DATASet と ISDAS 観測センサーの統計的性質に基づき，与えられたパケットのペイロードを見ることなくポートスキャンの種類を機械的に同定する決定木を発見した．主要な結論は，ポートスキャンのやり方は感染したマルウェアに依存しないということである．

Interrelation between Interactive and Non-interaction Sensors

Tomohiro Kobori† Hiroaki Kikuchi† Masato Terada††

†Graduate School of Engineering, Tokai University, 1117 Kitakaname, Hiratsuka, Kanagawa 259-1292
††Hitachi, Ltd. Hitachi Incident Response Team (HIRT), 890 Kashimada, Kawasaki, Kanagawa 212-8567

Abstract Our study aims to clarify if the port-scanning packets sent from bot-infected hosts are identical to the packets obserbed by distributed non-interactive sensors. Based on the statistical propeties of the CCC DATASet and the ISDAS, we show the decision trees to automatically identify port-scanning strategies, random or sequential without investigating the payload of given packets. The main result is that port-scanning behaviors are independent from the malware infected to the scanners.

1 はじめに

不正アクセスの主流は自立して無計画にポートスキャンを繰り返すワームから，連携して意図的な攻撃を制御可能なボットネットに移行したと言われている [2]．ワームの単純な感染の仕組みに対して，ボットはバッファオーバーフローを引き起こすローダー，マルウェア (MW) 本体をダウンロードする DL サーバ，IRC セッションを利用してスキャンなどのコマンドを送信するコマンドコントロール (C&C) サーバなどの多くのホストと複雑な通信を経て感染と攻撃に至る．それゆえ，スキャンアルゴリズムが事前にプログラミングされて固定していたボットとは異なり，C&C サーバからの命令によってスキャンパターンや攻撃対象が動的に変化することが予想される．すなわち，次の仮説が立てられる．

仮説 1 ポートスキャンは全てボットである．

仮説 2 ポートスキャンはダウンロードしたマルウェアの種類に依存しない．

さて，これらは本当だろうか？

そこで，本研究では研究用データセット “CCC DATASet 2008(以降，CCC2008 データ)” による研究用通信データを用いて従来の定点観測データと統合することにより，この仮説を検証する．本データセットでは，2 台のハニーポットによる 2 日間の全通信パケットをキャプチャーした攻撃通信データ (2pcap) と 112 台のハニーポットで観測されたマルウェア検体名が同定された攻撃元データ (3log) が提供されている．これらの解析を行いその統計的な性質を ISDAS[1] による受動的な定点観測データと比較することとで仮説 1 の立証を試みる．本データセットで同定された MW 名と実行される攻撃との間に相関がなければ仮説 2 が証明できる．しかしながら，観測データやシグネチャーの不完全性から次の問題点が生じる．

- MW の同定の困難さ．未知の検体に対して MW 名が同定できない．既知でも，ボットネット間で脆弱性のあるホストを奪い合い**多重感染**し，どの MW による影響か確定できない．
- C&C サーバの命令と攻撃の相関の不明確さ．命令があっても即攻撃をするわけではなく，**複数の命令**を受けることがあり攻撃と一意に結び付けられない．

そこで，キャプチャーデータから MW に固有である可能性のある様々な特徴量を抽出し，情報利得に基づく決定木学習にかけて，攻撃の種類（スキャンタイプ）と MW 検体の種類の識別に有益な属性を自動学習する解析方法を提案する．本稿では CCC2008 データに対して解析した結果と仮説の成立について報告する．

2 解析方法

2.1 タイムスロット

攻撃通信データは定期的にクリーンな状態にリセットするハニーポットが 2 日間に通信した全データのキャプチャーからなる．本稿では，Windows XP が送信する NTP パケットを利用して，攻撃通信データを観測単位区間となる**タイムスロット** 146 個に分割した．ここで，観測日 d からの連番 i の組 $d.i$ をタイムスロット ID とする．$i = 1$ から始まるので，例えば，$ID = 28.8$ は，4 月 28 日の 8 番目の観測単位区間を表す．

2.2 スキャンタイプ

タイムスロットにおける送受信パケット数の累積数を図 1 に示す．入力数と出力数が大きく違うので左右の 2 軸で表している．リセットしてから 1 秒後からパケットが届き始め，18 秒目にはバッファオーバーフローが起きて，MW のダウンロードが始まっている．複数の C&C サーバからの命令を受けて，747 秒後からポートスキャンを開始している．このように，LD, DL, C&C サーバとの通信は重複していて不確かだが，スキャンの開始は明確である．

IRC セッションの中で観測される `ipscan s.s.dcom2 -s` の命令文は，感染元の IP アドレスの下位 2 オクテットを 1 づつ増加させて行う**シーケンシャルスキャン** を意味する．例えば，タイムスロット 29.59 における宛先アドレスは図 2 に示すように，65536 周期で繰り返される．これを，スキャンパターン $ST = s_2$ と呼ぶ．同様にして，s_4, s_3 を定める．

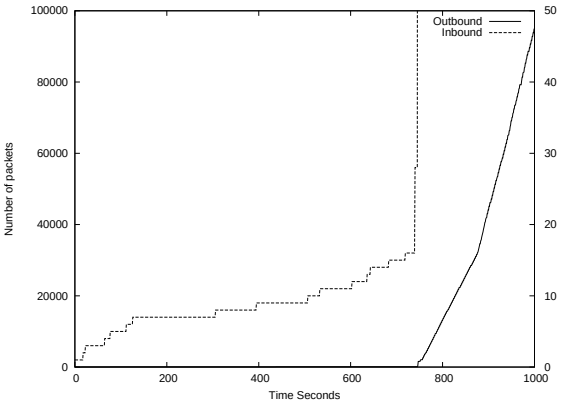

図 1: 累積パケット数（$i = 28.23$, 左 Y 軸 Outbound, 右 Y 軸 Inbound)

これに対して，宛先をランダムに決める走査を**ランダムスキャン**と呼び，$ST = rnd$ で表す．図 3 は，28.46 におけるランダムスキャンである．IRC の命令は，`advscan` や `asc` などが該当する．ただし，タイムスロットで観測される IRC セッションは単一とは限らない．そこで，可能性のあるスキャンパターンを (s_2, s_3, s_4, rnd) のビットパターンで表す．

スキャンを行う速度は感染ホストのパフォーマンスとスキャンパターンに依存する．単位時間にスキャンしたのべ宛先アドレス数を**スキャンレート**と呼び，SR [pkt/s] で表す．これは送信パケット数をスキャン期間で割って求める．シーケンシャルスキャンのレートが高く，一定であるのに対して，ランダムスキャンは分散が大きく低速である．

2.3 ポートに関する特徴量

攻撃通信データでは 135, 445，ICMP をこれを 3 ビットの P_D で表す．

一方，発信元ポート (以降，ソースポート）は通常の TCP では未使用のポートを順に割り当てるためにほとんどランダムに見える．しかし，通信攻撃データを観察してみると規則的なインクリメント (図 4) だけではなく，デクリメント (図 5) やランダム（図 6) の様な不規則性な振舞いをするものがあった．そこで，この差を MW を識別する特徴量として利用するために，時刻 t における**ソースポート差**

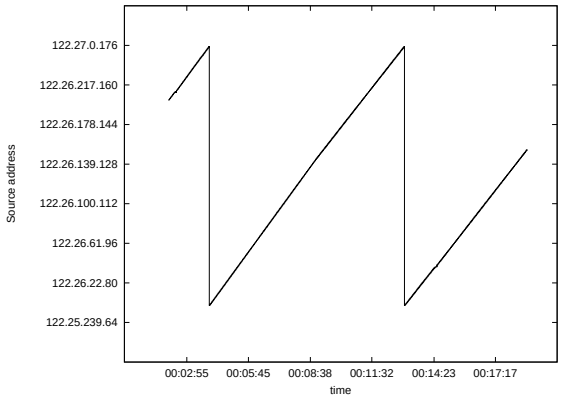

図 2: 宛先 IP の推移（シーケンシャル, $ST = s_2, i = 29.59$）

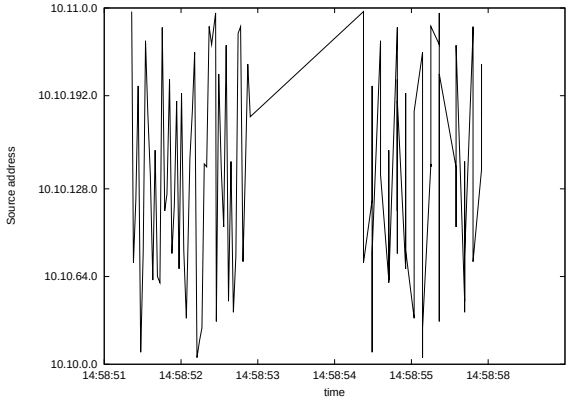

図 3: 宛先 IP の推移（ランダム, $ST = rnd, i = 28.46$）

分 $dP_S(t) = P_S(t) - P_S(t-1)$ を定義し，その平均値を d_{PS}, 標準偏差を v_{PS} で表す.

2.4 マルウェア識別名

マルウェアのダウンロードがあったかどうかは攻撃通信データより受信量の変化を見ることで判断できる．これを，0,1 の値を取る DL で表す.

ダウンロードされたマルウェアは，攻撃通信データから抽出したダウンロードの時刻と DL サーバのアドレスについて，CCC 2008 データの攻撃元データを検索することで識別できる．ただし，まだシグネチャーには加えられていない未発見の検体は Unknown と判断されるし，単一のタイムスロットに複数の DL があることもある．そこで，主要なマルウェアを (Pe, Worm, Bkdr, Troj) のタイプと (Bobax, Kolabc, Virut, Vanbot, Other) の名

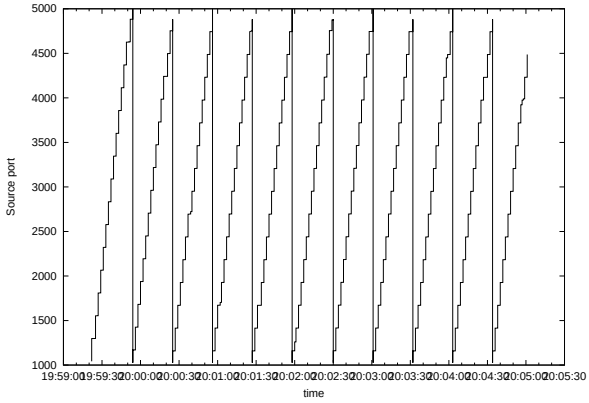

図 4: ソースポートの推移（インクリメント, $i = 29.61, ST = s_4$）

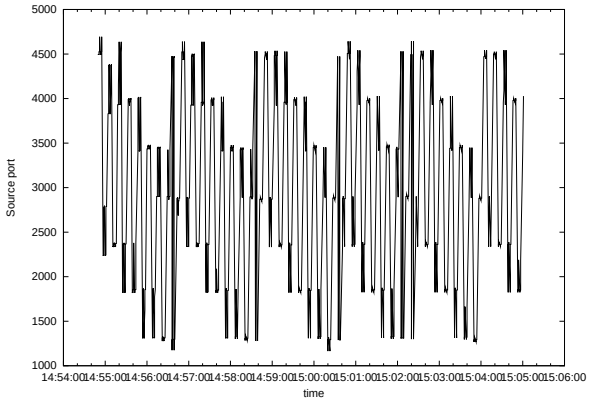

図 5: ソースポートの推移（デクリメント）

前の組合せで表し，その 9 ビットの値で表す．例えば，$MW = 100\,10100$ は Pe.Bobax, Pe.Virut の二つのマルウェアのダウンロードがあった事を表す.

2.5 特徴量抽出

表 1 に通信攻撃データから抽出する特徴量を示す.

ここで，ホストあたりの平均入出力パケット数は $H_{I/O} = C_{I/O}/U_{I/O}$ により定義する．例えば，マルウェアのダウンロードが起きていれば H_I が大きくなり，ポートスキャンを行っていれば C_O が大きくなるが，スキャン範囲が広いと H_O は小さい.

2.6 決定木学習

決定木は，ターゲット属性のエントロピーを最小化する最適な属性を貪欲的に選ぶことで，識別の決定木を学習するアルゴリズムである [3]．C4.5[4] は，

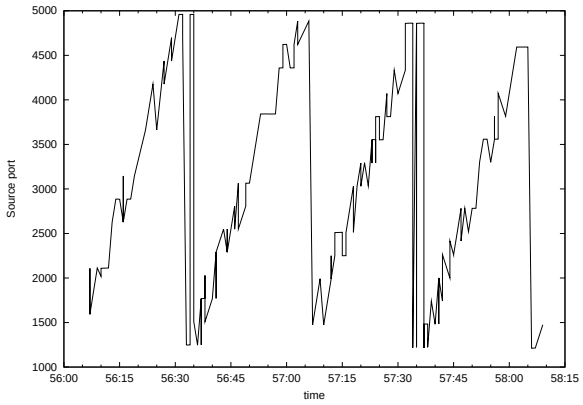

図 6: ソースポートの推移（ランダム）

表 1: 特徴量一覧

i	タイムスロット ID$(28.1, \ldots, 29.73)$
IP	ハニーポットアドレス
C_I, C_O	総入力 (出力) パケット数 [pkt]
DL	ダウンロードの有無 (0,1)
ST	スキャンタイプ (s_2, s_3, s_4, rnd)
DP	宛先ポート (135, 445, ICMP)
SR	スキャンレート [pkt/s]
d_{PS}	ソースポートの増分
v_{PS}	d_{SP} の標準偏差
U_I, U_O	ユニーク発信元 (宛先) アドレス数
H_I, H_O	ホストあたり平均入力 (出力) パケット数
MW	マルウェア名 (Pe, Worm, Bkdr, Troj, Bobax, Kolabc, Virut, Vanbot, Other)

学習データに出現する値を総当りすることで，連続値の閾値も対応している．

特徴量に顕在する不確定性，すなわち，1) スキャンパターンの不確定性，2）マルウェア識別名の不確定性に対処するため，$n = 146$ スロットのデータから確実に判定が可能な部分集合を選びそれを学習データとして決定木を作る．ひとたび決定木が出来れば，不確定のスロットも機械的に識別できるからである．学習データには，(0) 未感染（判別が容易）のスロット $n_0 = 58$ (1) 単一の命令のみのスロット $n_1 = 53$, (2) 単一かつ既知 MW 名が識別されているスロット $n_2 = 53$ を用いた．

3 解析結果

3.1 タイムスロットと特徴量

抽出した特徴量および平均値を表 4（4 月 28 日分）に示す．各スキャンタイプで平均した特徴量を表 2 に示す．スキャン頻度 SR は 167 から 250 に分布しているが，分散が大きすぎて，スキャンタイプを識別するには十分な差は得られない．スキャンタイプはむしろホスト当たりの送信パケット数 H_O に現われている．s_2 のみ他より 3 倍程度多い．

表 2: スキャンタイプについての統計値

ST		C_I	SR [pct/s]	v_{PS}	H_O [pct/addr]
s_2	μ	3401.9	194.5	0.025	10.7
	σ	2051.9	59.9	0.033	17.9
s_3	μ	6014.5	250.9	0.013	2.8
	σ	1011.8	112.8	0.009	0.7
s_4	μ	1850.0	167.6	0.017	3.7
	σ	1850.1	63.0	0.044	5.3
rnd	μ	1635.6	140.0	0.125	3.7
	σ	1271.6	97.0	0.167	1.1

各マルウェアの平均の統計量を表 3 に示す．Virut は単位ホスト当たりに対するパケット送信量が多い．これは第 3, 第 4 オクテットスキャンを行わない s_2 の影響であると考えられる．

3.2 スキャンパターンの決定木

スキャンパターンのあいまいさのない学習データとして，2.6 節で述べた $n_0 + n_1 = 101$ 個を用いて学習した木を図 7 に示す．

ここで，木のノードは特徴量（属性），枝はその値を示す．リーフは目的属性であり，(識別される事例数/誤識別数) が表示されている．例えば，ルートから下に木の左端をたどる経路は，スキャンパターンの評価値 STE を決める

$$C_O > 2065 \wedge C_I > 1487 \wedge U_O \geq 73762 \Rightarrow STE = s_2$$

というルールを表している．この条件にあうスロットは 9 個あるが，内 2 個は誤っている．なお，u は未感染のパケットを示している．スキャンを行っているのは，送信数が多い時であり，送信先の種類が多いものが s_4, 少ないものが s_2, ポート 135 を使わないものが rnd であることを意味している．

図 7 より, スキャンタイプ ST は, 総パケット送信数 C_O，総受信数 C_I，ユニーク送信アドレス数 U_O とで十分正確に分類出来ることが示された．図 8 に，宛先ホストあたりの平均送信量 H_O と総パケット送信数 C_O についての散布図を示す．決定木の閾値で識別が正確に行われていることが図示されている．

決定木の精度を示すために，IRC のコマンドから判断したスキャンタイプ ST と決定木による評価値

表 3: MW についての統計値

MW 名	H_I [pct/addr]	H_O [pct/addr]	P_D	SR [pct/s]	d_{PS}	
					μ	σ
BOBAX	58.72	26.8	135,445	89.23	0.022	0.031
KOLABC	34.01	14.85	135,445	97.78	0.036	0.08
VIRUT	43.09	28.08	135,445	149.16	0.016	0.029
VANBOT	33.01	7.5	135	124.78	0.027	0.07

表 4: 攻撃通信データのタイムスロット特徴量（部分）

i	C_I	C_O [pkt]	DL	ST	P_D [pct/s]	SR	v_{PS}	U_I	U_O [addr]	H_I [pkt/addr]	H_O	MW
28.1	1	25	0	0000	000	0	0	1	3	1	8.3	000000000
28.2	6	133	0	0000	000	0	0	6	8	1	16.6	000000000
28.3	151	331	0	0000	100	0	0	9	11	16.8	30.1	000000000
28.4	113	297	0	0000	100	0	0	14	15	8.1	19.8	000000000
28.5	1487	304707	1	0010	000	210	0.08	23	65558	64.7	4.6	100010100
28.6	1058	232700	1	0010	000	122.9	0.02	19	115477	55.7	2	100000100
28.7	89	255	0	0000	100	0	0	12	13	7.4	19.6	000000000
28.8	104	275	0	0000	000	0	0	12	12	8.7	22.9	000000000
28.9	239	11201	1	0010	000	130.3	0	33	5410	7.2	2.1	000100001
28.10	300	619	0	0000	000	0	0	133	134	2.3	4.6	000000000
28.11	168	425	0	0000	000	0	0	161	156	1	2.7	000000000
28.12	200	487	0	0000	100	0	0	139	140	1.4	3.5	000000000
28.13	78	241	0	0000	000	0	0	10	11	7.8	21.9	000000000
28.14	386	250641	1	0010	000	231	0.01	20	65305	19.3	3.8	000100001
28.15	146	363	0	0000	100	0	0	10	11	14.6	33	000000000
28.16	86	525	0	0000	000	0	0	5	6	17.2	87.5	000000000
28.17	5298	327958	1	1010	000	234.8	0.13	62	65607	85.5	5	100010100
28.18	46	187	0	0000	000	0	0	9	10	5.1	18.7	000000000
28.19	112	229	0	0000	000	0	0	6	7	18.7	32.7	000000000
28.20	4	131	0	0000	000	0	0	3	4	1.3	32.8	000000000
28.21	41	161	0	0000	100	0	0	4	5	10.3	32.2	000000000
28.22	110	349	0	0000	000	0	0	10	11	11	31.7	000000000
28.23	6262	157704	1	1000	100	200.1	0.07	65	44357	96.3	3.6	100000100
28.24	225	473	0	0000	100	0	0	17	18	13.2	26.3	000000000
28.25	1850	351467	1	0010	100	235.1	0.01	18	89885	102.8	3.9	110011100
28.26	6109	302193	1	1000	000	213.8	0.1	80	65623	76.4	4.6	111011010
28.27	4683	593957	1	1010	100	288.4	0.07	70	113707	66.9	5.2	100010010
28.28	182	399	0	0000	100	0	0	7	8	26	49.9	000000000
28.29	3971	518940	1	1010	000	304.4	0.04	58	99915	68.5	5.2	100010100
28.30	368	357531	1	0010	000	240.8	0	17	91927	21.6	3.9	100000100
平均	1188	115152	–	–	–	124.8	0.0146	58.0	37912	35.24	16.17	–

表 5: スキャンタイプの決定木の精度

ST\ 評価値	rnd	s_2	s_4	u	total
未分類 m	2	5	24	1	32
rnd	2				2
s_2		7		4	11
s_4		2	38	1	41
u	1			57	58
total	5	14	62	63	144
適合率 P_{ST}	.67	.78	1.0	.92	.93

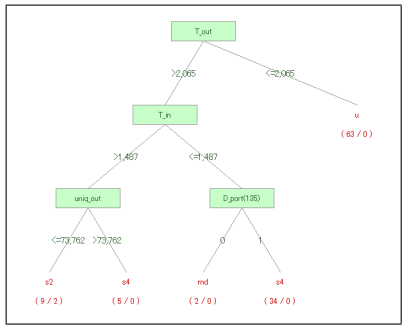

図 7: C4.5 によるスキャンタイプの決定木

STE との比較を表 5 に整理した．表の対角線に分類されるスロットが多いのは，決定木が正確に識別できることを示している．m(多重命令による未分類)を除くデータに対する精度を，再現率 R_{STE} と平均適合率 P_{STE} で評価すると，次のようになる．

$$
\begin{aligned}
R_{STE} &= \text{正識別スロット数/対象スロット数} \\
&= 104/112 = 0.93 \\
P_{STE} &= \text{正識別スロット数/識別スロット数} \\
&= E\left[\frac{|\{i|ST_i = STE_i = j\}|}{|\{i|STE_i = j\}|}\right] \\
&= (3 \cdot 0.67 + 9 \cdot 0.78 + 38 \cdot 1 + 62 \cdot 0.92)/112 \\
&= 0.93
\end{aligned}
$$

3.3 マルウェアの決定木

同様に，マルウェア識別が確定的な学習データとして，$n_2 = 53$ 個のスロットを用いた決定木を 9 に示す．枝狩りによる簡単化により，母数の少ない BOBAX が省略されている．この決定木によって分類される MW 名を MWE で表す．例えば，条件

$$C_I > 589 \vee H_O > 2 \wedge (U_I > 11 \vee H_I > 30.7)$$

が成立する時，$MWE =$ VIRUT である．

C4.5 によって得られた属性によって作成した散布図を図 10 に示す．閾値 $C_I = 589$, $H_I = 21$ によっ

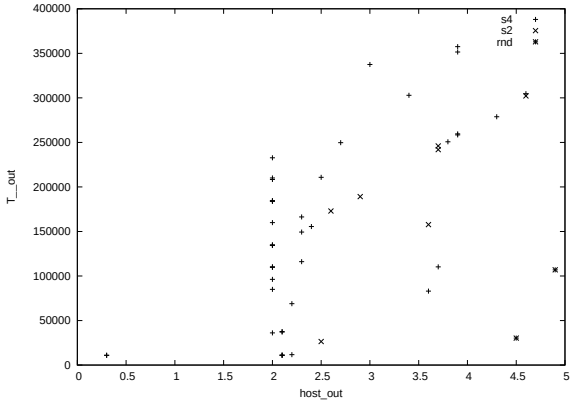

図 8: スキャンタイプの散布図

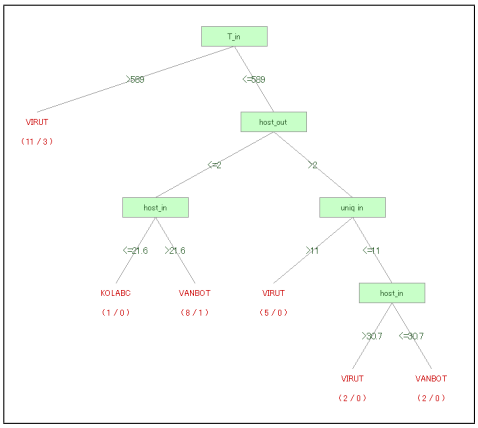

図 9: C4.5 による MW の決定木

て，領域が VANBOT と VIRUT に 2 分割されている．

識別の精度を確かめるために，攻撃元データのシグネチャーによるマルウェア名 MW と決定木の評価 MWE を比較したスロット数を表 6 で整理した．ここで，u は未感染，MW = UNKNOWN は未知の検体と複数の検体が検出されて同定できないスロットを表している．Unknown 以外の 91 スロットに対する再現率 R_{MWE} と適合率 P_{MWE} は次の通りである．

$$\begin{aligned} R_{MWE} &= 72/91 = 0.79 \\ P_{MWE} &= 0.69 \end{aligned}$$

4 考察

4.1 マルウェアとスキャンタイプの相関

仮説 2 マルウェアとスキャンは独立だろうか．

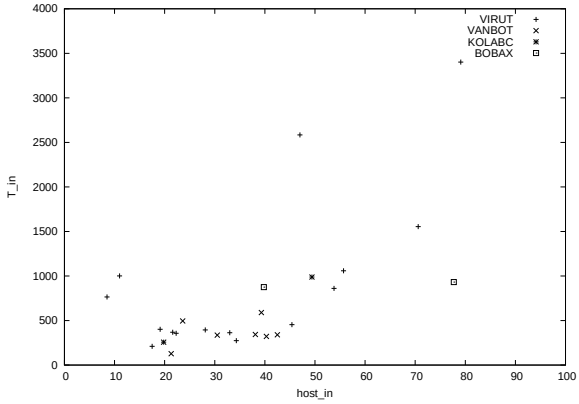

図 10: C4.5 による MW の散布図

表 6: MW の決定木の精度

$MW\backslash$ 評価値	u	KO.	VB.	VR.	total
UNKNOWN	6		1	47	54
u	53		1	4	58
BOBAX					2
KOLABC		1		2	3
VANBOT			3	5	9
VIRUT	4			15	19
適合率 P_{MWE}	.91	1.0	.75	.2	.69

まず，重複や不確定性を無視して，攻撃通信データの命令から抽出したスキャンパターン ST と攻撃元データから判別したマルウェア識別名 MW の相関を表 7 に示す．どの MW に感染してもスキャンパターンは一様であり，ほとんど独立のように見える．例えば，BOBAX に感染して s_2 のスキャンを行う同時確率は次のように独立である．

$$\begin{aligned} Pr[MW = B_{OBAX}, ST = s_2] &= 17/219 = 0.078 \\ &= Pr[MW = B_{OBAX}]Pr[ST = s_2] \\ &= 0.32 \cdot 0.23 = 0.072 \end{aligned}$$

それぞれの確率の積で近似できる．MW 名の不確定性の影響も考えられるが，決定木を用いて予測した MWE と STE の比較をしてもほぼ同様に確率的に独立であることを示していた．従って，以上の帰着より仮説 2 は成立すると結論付ける．

表 7: マルウェアとのスキャンタイプ比率

$MW\backslash ST$	s_2	s_3	s_4	rnd	total
BOBAX	17	2	28	3	35
KOLABC	7	2	14	5	18
VIRUT	30	4	42	5	58
VANBOT	7	0	20	1	25
Other	8	2	20	2	32

表 8: ポートスキャンパケットの平均到着間隔

	CCC 2008	ISDAS
n	76 slot	50 sensor
$\mu(T)$	140.25	157.67
$\sigma(T)$	183.38	0.0465

表 9: 宛先ポートの分布の比較

P_D	ISDAS [pkt]	[%]	CCC [slot]	[%]
135	697488	41	76	82
445	158616	9	13	14
ICMP	346447	20	3	3
Other	500791	29	0	0

4.2 定点観測との比較

4.2.1 平均到着間隔

CCC2008 データと ISDAS でのポートスキャンパケットの平均到着間隔 T に着目する．到着間隔は攻撃の頻度を表しており，同じ期間 (2007 年 11 月 1 日〜2008 年 4 月 30 日) に観測された頻度は一致するはずである．ただし，CCC 2008 データにはダウンロードなどの副作用が多いので，タイムスロットの開始から最初に脆弱性をついたパケットが届くまでの時間を用いた．146 スロットのうち感染したスロットは 76 であり，その平均を表 8 に示す．

表に示されるように，非常に近い間隔であった．両者の間には ISDAS の平均間隔の 11%の差しかなく，CCC 2008 データの観測数が少なく分散が大きいことを考慮すれば，ほぼ同じ頻度で攻撃が起きている．

4.2.2 宛先ポート分布

攻撃の種類を比較するために，スキャンの宛先ポートに着目する．ISDAS で観測される全パケットの宛先ポートの割合と CCC 2008 データの攻撃通信データのスキャンを行った 83 スロットのおける宛先ポートについての割合を表 9 に示す．ポートの第 2 位と 3 位の順位が入れ替わり，それ以外のポートのスキャンも行われていないなど，相違が生じている．観測期間の不足やセンサーの数の違いが相違の原因か，仮説の不成立である可能性もある．

解析で算出したウィルス毎のスキャンパターンの比率を用いて攻撃元データで Pe.Virut と判定されたログデータを抽出し, その確率を適応した．その結果を, 図 11 に示す．s_2 と s_4 の確率が同じために

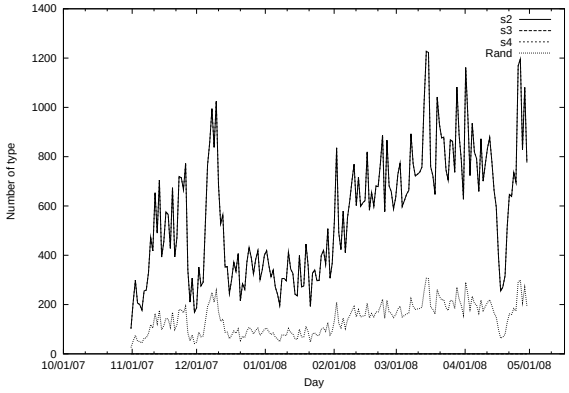

図 11: 過去半年間の Virut におけるスキャンタイプ数

グラフでは 2 つの属性に見えてしまうが, 実際には 3 つの属性が存在している．

4.2.3 スキャン速度

二つの観測データのスキャンレート SR [addr/s] を比較した結果を代表的な 2 種類のワームと共に，表 10 に示す．

ここで，CCC 2008 データは攻撃通信データのスキャンタイプの平均値，ワームは, 検体を感染させて観測したスキャン速度を用いた．ISDAS においては, センサーのアドレスが観測中に変動してしまう可能性もあり，ペイロードも見えないのでスキャンタイプの同定も困難である．そこで，同一のスキャン間隔で複数回観測しているセンサーを目視により検出し, s_2 のスキャンをしていると仮定して観測時間差からスキャン速度を同定した．

残念ながら，CCC 2008 データのスキャンレートと一致する ISDAS データは見つけられなかった．表に示すように，周期的な振舞いをしていても，その速度は大きく異なり，同一のコードから攻撃されているとは言えない．スキャンレートはマルウェアだけでなく，感染したホストのパフォーマンスにも大きく依存する．あるいは，ISDAS のスキャンが s_2 であるとした仮定が誤っているかもしれない．興味深いことに，参考のために示したワームのスキャン速度は CCC 2008 データよりも ISDAS に近い．

表 10: 3 つ観測データによるスキャンレート SR

Dataset	ID	SR [addr/s]
CCC2008	$ST = rnd$	23.78
	$ST = s_2$	148.44
	$ST = s_4$	167.96
ISDAS	12/29/07 6:28:34	29.09
	2/21/08 2:45:09	9.13
	2/25/08 13:33:54	42.09
ワーム	Blaster	11.08
	Dasher	4968.61

5 結論

CCC 2008 データ攻撃通信データと攻撃元データを解析して，定点観測データ ISDAS との比較を行った．結論を次に示す．

(1) 仮説 1「ポートスキャンは全てボットによって行われている」は成立しない．平均到着間隔（スキャン密度）は 11%の誤差で一致が見られたが，宛先ポートの分布とスキャン速度の観点では，定点観測センサーで観測しているものはボットによる攻撃ではない．

(2) 仮説 2「ポートスキャンのタイプはマルウェアに依存しない」ことは，攻撃通信データから求めた確率密度により裏づけされた．すなわち，マルウェアは様々なスキャンアルゴリズムを実装しており，C&C サーバからの命令だけに依存して攻撃を行っている．

(3) 決定木学習により，ペイロードを見ることなく通信パターンの特徴量に基づいて，再現率 93%, 適合率 94%の精度でスキャンパターンの同定が可能である．識別には，総送信パケット数とユニーク宛先アドレス数が有益である．

(4) ペイロードを見ることなく通信パターンの特徴量に基づいて，再現率 79%, 適合率 69%の確からしさでマルウェア名の同定が可能である．識別に有意な特徴量は，パケット数やユニーク発信元アドレス数である．

謝辞

本研究の遂行にあたって，定点観測データを提供いただいた JPCERT/CC と CCC に感謝する．本研究に有益なご助言を頂いた（株）日立製作所の仲小路博史氏，鬼頭哲郎氏，藤原将志氏に感謝する．

参考文献

[1] 戸田，他，“ISDAS: Internet Scan Data Acquisition System”, 情報処理学会，コンピュータセキュリティシンポジウム CSS2004，pp. 199-204，2004.

[2] 阿部，田中，“C&C セッション分類によるボットネットの検出手法の一検討”, 第 6 回情報科学技術フォーラム (FIT2007), pp. 77-78, 2007.

[3] Tom Mitchell, Machine Learning, McGraw Hill, 1997.

[4] Quinlan, J. R., “C4.5 Programs for Machine Learning”, Morgan Kaufmann, San Mateo, California, 1992.