東海大学大学院2011年度 修士論文

Apriori とPrefixSpan による
連携感染のハイブリッド検出方式

Apriori–PrefixSpan Hybrid Approach for
Automated Detection of
Botnet Coordinated Attacks

指導教員 菊池 浩明 教授

東海大学大学院 工学研究科 情報理工学専攻

0BDRM004 大類 将之

目 次

第1章　序論

1.1　背景

過去 10 年間において，マルウェアによる脅威は増加傾向にあり，非常に巧妙化している [1]．特に近年，インターネット上で感染が拡大している不正プログラムの一種**ボット**には非常に多くの亜種が存在し，複数のダウンロードサーバに分散して感染するなど，複雑化，高度化が進んでいる [2].

ボットに感染したコンピュータは，悪意あるユーザ (以下，攻撃者) が用意した指令サーバなどに自動的に接続され，数十〜数百万台規模のボット感染コンピュータを従えた**ボットネット**と呼ばれる巨大ネットワークを形成する [3, 4, 5]．具体的には，自己複製するワームや電子メール，Web サイトなど様々な手法を用いてマルウェアを送信し，更なるコンピュータを感染させることで，自身を拡大する．

また，ボットネットは攻撃者が任意のコマンドを送信することで制御が可能であり，様々な不正な目的に使用されている．例えば，分散型サービス使用不能攻撃 (DDoS 攻撃) や大量のスパムメール送信，不正クリック詐欺，脆弱なサーバ情報やクレジットカード番号の収集などが挙げられる．その有用性ゆえに，ボットネットは貸し出しも行われており，ボットネットのビジネス化も問題視されている [6]．ボットネットのメカニズムとしては，中央集中型と P2P 型の 2 種類があり，中央集中型では，ボットは主要な Command & Control サーバ (以下，C&C) と通信を行う．多くのボットネットは C&C サーバに Internet Relay Chat (IRC) を使用しており，攻撃者は C&C サーバを経由してボットネットにコマンドを送信する．P2P 型は Peer-to-Peer による分散制御のメカニズムを用いる．

幸いなことに，ボットネットの活動は，ネットワーク上のマルウェアを観察することによって，追跡することが可能である．この方法は「ハニーポット」システムで使用される．例えば，McCarty は、ネットワーク内のコンピュータへの攻撃に関する有用なデータをキャプチャするハニーポットの実装の例を示している [7]．ハニーポットとは，攻撃者にとって魅力的に見える脆弱なコンピュータのように振る舞うおとりホスト，すなわち，攻撃を受けることを専用としたホストである [8].

本論文では，ハニーポットで収集した研究用データセット CCC DATAset[10] を使用し，中央集中型ボットネットに焦点をあて，マルウェアの解析を行う[1)].

[1)]CCC DATAset については第 3 章にて後述する．

1.2 目的

最大の問題は，ボットネットによる攻撃や感染活動が限定的かつ水面下で実施される点にある．ユーザ自身が攻撃や感染の事実を把握できないだけでなく，自身が加害者になり得るという深刻な状況であるが，対策は難しい．例えば，ボットにおける感染に関して，複数のサーバを連携して感染する特徴が報告されている [9]．表 1.1 は，CCC DATAset 2009 攻撃通信データの中から，PE_VIRUT.AV に感染したとき，数分後に TROJ_BUZUS.AGB と WORM_SWTYMLAI.CD が同時に感染している例である．

表 1.1: 攻撃通信データから抽出した MW の連携活動例

時刻	DL ホスト IP アドレス	Dst Port	プロトコル	MW 名
0:02:11	124.86.***.111	47556	TCP	PE_VIRUT.AV
0:03:48	67.215.*.206	80	TCP	TROJ_BUZUS.AGB
0:03:48	72.10.***.195	80	TCP	WORM_SWTYMLAI.CD
0:36:46	124.86.**.109	33258	TCP	PE_VIRUT.AV
0:36:52	72.10.***.195	80	TCP	WORM_SWTYMLAI.CD
0:36:52	67.215.*.206	80	TCP	TROJ_BUZUS.AGB
0:46:56	124.86.**.109	33258	TCP	PE_VIRUT.AV
0:48:52	67.215.*.206	80	TCP	TROJ_BUZUS.AGB
0:48:52	72.10.***.195	80	TCP	WORM_SWTYMLAI.CD

これより，ダウンロードサーバが異なる IP アドレスであっても，マルウェアの感染活動には類似性があることがわかる．このように，複数のダウンロードサーバが連携し，個別のマルウェアを組み合わせて感染させる攻撃を**連携感染**と定義する．

連携感染の一連の流れを以下に示す．

1. PE_VIRUT.AV に感染 [124.86.165.*]
2. ss.ka***.com (DNS) を名前解決し，hub.56***.com (IRC) に接続 [67.43.226.*]
3. always***.com (DNS) [67.215.1.*]，zonet***.info (DNS) [72.10.166.*] を名前解決
4. 各サーバから TROJ_BUZUS.AGB (/vot.exe) [67.215.1.*]，WORM_SWTYMLAI.CD (/vss.exe) [72.10.166.*] を HTTP GET によりダウンロードする
5. ポートスキャンを行う

すなわち，起点となるマルウェアをダウンロード後，IRC サーバに接続し，他のマルウェアを HTTP GET によりダウンロードする．連携感染によりサーバの数や感染するマルウェアの数など若干の違いはあるが，概ね同様である

しかし，このような連携活動を発見するためには，多量のデータの中から共通に生じるパターンを抽出する必要があり，非常に困難である．例えば，CCC DATAset 2009 攻撃元デー

タを対象に，表 1.2 を元にした出現数トップ 4 のダウンロードホスト IP アドレスのダウンロード数の推移を示した図 1.1 について考えよう．

表 1.2: 攻撃元データから抽出した DL ホスト IP アドレス上位 10 個のデータ

順位	DL ホスト IP アドレス	DL 回数	平均 DL 回数	MW 数	ハニーポット数
TOP1	72.10.***.74	462246	3884.4	119	91
TOP2	72.10.***.195	399562	8324.2	48	92
TOP3	85.114.***.2	33283	1147.7	29	82
TOP4	85.114.***.207	32202	870.3	37	78
TOP5	67.215.*.206	26780	3825.7	7	59
TOP6	211.95.**.6	19641	198.4	99	85
TOP7	72.10.***.26	14951	287.5	52	82
TOP8	92.48.**.63	11699	117.0	100	69
TOP9	67.18.***.250	10060	76.8	131	68
TOP10	72.8.***.164	5099	127.5	40	81

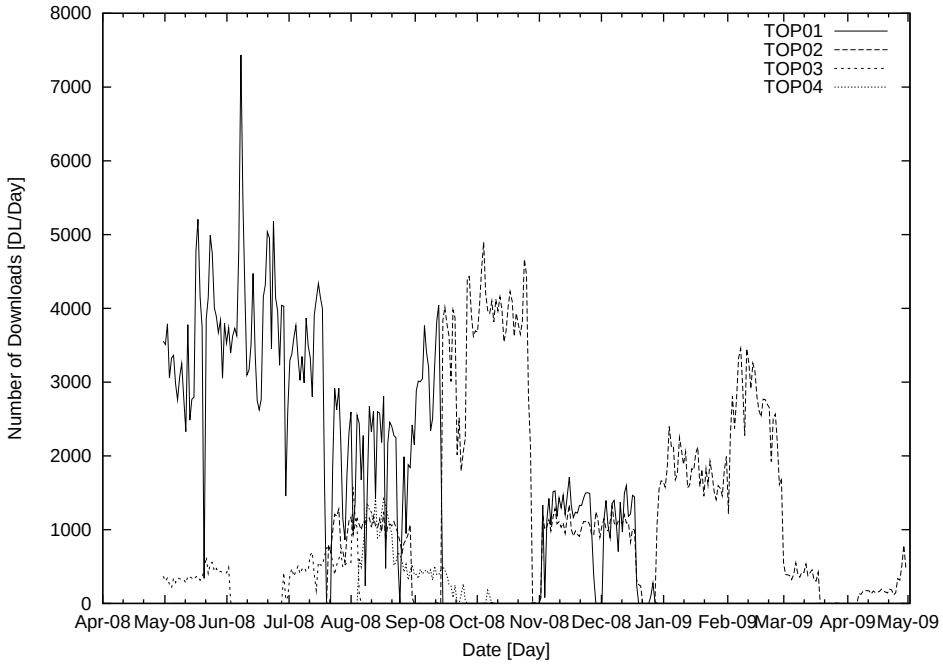


図 1.1: 1 年間で観測されたダウンロード数の推移

IP アドレスや時期によるダウンロード数の変動は大きい．全く観測しない期間も存在し，ハニーポットによっても結果は異なる．そのため，連携感染だと考えられる特徴を発見しても，その真偽を測るのは難しい．1 年間で 1,335 種類の MW が観測され，先の例のように 3 種類の異なる連携を考えるだけでも，${}_{1,335}C_3 = 395,654,395$ 個組み合わせがある．しかも，365 日/年 × 94 台 × 24 時間/日 × 3 スロット/時間 = 2,470,320 の全てのタイムスロットにつ

いてそれらが成立するかを総当りで調べるのは現実的ではない[2)].

そこで，本論文では，以下の二つのデータマイニング手法を適用し，大規模データから頻出する共通の攻撃パターン，すなわち，連携感染を効率的に検出する手法を提案する．

1. *Apriori* **アルゴリズム**
   Apriori アルゴリズムとは，Agrawal らが提案した代表的な相関ルール抽出アルゴリズムである [11]．支持度と確信度という閾値に最小値を与え，その値を元に数多く抽出される相関ルールの中から，効果的に価値の低いルールを枝刈りする．これにより，価値の低いルールを除き，効率よく価値あるルールを発見できる．

2. *PrefixSpan* **アルゴリズム**
   PrefixSpan アルゴリズムとは，Pei らが提案した Prefix Projection という射影を用いた系列パターンマイニングのアルゴリズムである [12]．系列データから，射影対象の系列より後ろに存在するアイテムのみを抽出し，深さ優先で射影を繰り返すことで，頻出する系列パターンを効果的に発見する．

また，その結果明らかになった特徴を報告すると共に，2 つのアプローチを組み合わせたハイブリット検出手法を提案する．

1.3 論文構成

本論文の構成は次の通りである．

まず，第 2 章で Apriori と PrefixSpan のアルゴリズムについて，第 3 章で実験データである CCC DATAset について説明する．次に実験として，第 4 章で Apriori，第 5 章で PrefixSpan をそれぞれ適用して連携感染パターンを抽出した結果や考察を，第 6 章で過去 3 年間のデータを用いた連携感染の変遷を報告する．最後に，第 7 章でハイブリッド検出手法を提案し，第 8 章で関連研究と本研究を応用できる可能性を示し，第 9 章で結論と今後の課題を述べる．

[2)] タイムスロットの定義は第 3 章にて後述する．

第2章
要素技術

2.1 *Apriori* アルゴリズム

2.1.1 相関ルール

アソシエーション分析とは，データベースの中から X(前件部) $\Rightarrow$ Y(結論部) という相関ルールを抽出するデータマイニング手法である．

支持度とは，ルールの出現率を表し，全トランザクション N のうち，ルールの条件 X と結論 Y を共に含む確率

$$Supp(X \Rightarrow Y) = \frac{X \cap Y}{N}$$

で定義する．

確信度とは，ルールの関連性の強さを表し，ルールの条件 X が発生するトランザクションのうち，条件 X と結論 Y を共に含む確率

$$Conf(X \Rightarrow Y) = \frac{X \cap Y}{X}$$

で与える．

また，ルールの価値を測る手法として，**リフト値**があり，

$$Lift(X \Rightarrow Y) = \frac{Conf(X \Rightarrow Y)}{Y}$$

で定義する．

ルールの条件 X を前提とした場合と，していない場合の確信度の比率を表し,「結論部 Y が出現した割合」に対して,「前件部 X を 1 にしたとき，Y が同時に出現する割合」が何倍かを示している．すなわち，リフト値が 1 より大きい場合，ルールの条件 X を前提とした場合の方が結論部 Y が出現する確率が高い．

例えば，$X \Rightarrow Y$ の確信度が 0.8 であっても,「結論部 Y の出現した割合」が同じく 0.8 かそれ以上の場合は，Y の出現率が元から 80 %以上であるため，条件 X は Y に影響を与えない．しかし，逆に 0.8 未満の場合，Y の出現率より「X と Y が同時に出現する割合」が高いため，リフト値は 1 より大きい値を示し，$X \Rightarrow Y$ の関連性は高いと判断できる．

なお，本研究では支持度と確信度を使用し，リフト値は使用しない．

2.1.2 Apriori アルゴリズム

Apriori アルゴリズムとは，Agrawal らが提案した代表的な相関ルール抽出アルゴリズムである [11]．支持度と確信度という閾値に最小値を与え，その値を元に数多く抽出される相関ルールの中から，効果的に価値の低いルールを枝刈りする．これにより，価値の低いルールを除き，効率よく価値あるルールを発見できる．

表 2.1 で与えるトランザクションデータを用いて，$B, C \Rightarrow E$ の相関ルールの評価例を示す．

表 2.1: トランザクションの例

TID	A	B	C	D	E
1	1		1	1	
2		1	1		1
3	1	1	1		1
4		1			1

$$Supp(B, C \Rightarrow E) = 2/4 = 0.5$$

$$Conf(B, C \Rightarrow E) = 2/2 = 1$$

$$Lift(B, C \Rightarrow E) = \frac{2/2 = 1}{3/4 = 0.75} = 1.33$$

これから，$B, C \Rightarrow E$ のルールは支持度 50 %，確信度 100 %であることがわかる．つまり，このルールは 50 %の確率で出現し，B, C が発生した場合，100 %の確率で E が発生することを示している．同様に，リフト値も 1.33 なので，E が現れる確率よりも，$B, C \Rightarrow E$ が現れる確率の方が高いことを示している．ただし，この例では確信度が 1 であるために，リフト値も 1 以下になることがない．すなわち，価値を測る指標として参考にならないことに注意したい．

2.2 *PrefixSpan* アルゴリズム

2.2.1 系列パターンマイニング

系列パターンマイニングは，Agrawal と Srikant が提案したデータベース内から系列パターンを発見できる手法である [13]．シーケンスの集合に対し，ユーザは条件として最小支持度という閾値を与える．ここで，各シーケンスは要素のリストであり，各要素はアイテムの集合である．系列パターンマイニングは最小支持度を元に，全ての頻出するサブシーケンスを発見する．このとき，サブシークエンスの発生頻度は，最小支持度より大きいか等しい値となる．

2.2.2 PrefixSpan アルゴリズム

PrefixSpan(Prefix-projected sequential pattern mining) アルゴリズムとは，Pei らが提案した Prefix-projection という射影を用いた系列パターンマイニングのアルゴリズムである [12]．系列データから，射影対象の系列より後ろに存在するアイテムのみを抽出し，深さ優先で射影を繰り返すことで，頻出する系列パターンを効果的に発見する．PrefixSpan は，単に短い頻出する系列から，より長い頻出する系列を再帰的に探索するため，候補集合を生成する必要がなく，低メモリ要求で系列パターンを抽出できる [14].

まず，a_i，b_j をアイテム，α_i,β_j をアイテムのシーケンスとし，$\alpha = \langle a_1 a_2 \cdots a_n \rangle$，$\beta = \langle b_1 b_2 \cdots b_m \rangle$ とする．このとき，α は β の**サブシーケンス**であり，$\alpha \sqsubseteq \beta$ と表す．また，整数 $j_1, j_2, \cdots, j_n$ が存在するとき，$1 \leq j_1 < j_2 < \cdots < j_n \leq m$ とし，$a_1 = b_{j_1}, a_2 = b_{j_2}, \cdots, a_n = b_{j_n}$ とする．

次に，シーケンスデータベース S は，タプル $\langle sid, s \rangle$ の集合であり，sid は**シーケンス ID**，s は**シーケンス**を表す．データベース S 内のシーケンス α の**支持度 (support)** は，α を含むデータベースのタプルの数とする．すなわち，$support(\alpha) = |\{\langle sid, s \rangle | \langle sid, s \rangle \in S, \alpha \sqsubseteq s\}|$ である．また，閾値として正の整数の支持度 *min_sup* を与えたとき，シーケンス α にデータベース S 内の最小支持度 *min_sup* のタプルが含まれる場合，そのシーケンスを**頻出系列パターン**と呼ぶ．すなわち，$support(\alpha) \geq min\_sup$ である．さらに，シーケンス内のアイテム数は，シーケンスの長さ **length** と定義し，長さ ℓ の系列パターンを ℓ **系列パターン**と定める．

PrefixSpan アルゴリズムを以下に示す．α, β は，それぞれシーケンス $\langle a_1 \cdots a_n \rangle$, $\langle b_1 \cdots b_m \rangle$ とする．

1. **Prefix(接頭辞)** と **Postfix(接尾辞)** : $a_i = b_i$ for $i = 1, \cdots, m$ のとき，シーケンス α は β の Prefix となる．例えば，$\langle a\ a\ b\ c\ d\ d\ a\ b \rangle$ があるとき，Prefix は $\langle a\ a\ b\ c \rangle$ であり，Prefix に続く残りのシーケンス $\langle d\ d\ a\ b \rangle$ が Postfix である．

2. **Projection(射影)** : α, β, γ をシーケンスとし，$\beta \sqsubseteq \alpha, \gamma \sqsubseteq \alpha$ とする．次の 2 点が成り立つとき，シーケンス γ は，α の β-**projection** とする．(1) β が γ の Prefix である，(2) β が Prefix であるとき，α のサブシーケンスが存在しない．例えば，$\langle a\ a\ b\ \mathbf{c}\ d\ c\ d\ a\ b \rangle$ の **c**-projection は，$\langle d\ c\ d\ a\ b \rangle$ となる．

シーケンスデータベース S の例を表 2.2 に示す．表 2.2 を元に，ユーザが最小支持度 $min\_sup = 2$ を指定したときの PrefixSpan アルゴリズムの流れを以下に示す．

表 2.2: A sequence database

Sequence id	Sequence
100	PE WO TR
200	PE TR WO
300	BK PE TR TS WO
400	TS PE PE TR WO BK
500	PE WO TR WO

Step 1: Find 1-pattern sequences.

データベースを S をスキャンし，シーケンス内の全ての頻出する 1 つのアイテムを発見する．最小支持度 $min\_sup = 2$ なので，表 2.2 から，⟨PE⟩:5，⟨WO⟩:5，⟨TR⟩:5，⟨BK⟩:2，⟨TS⟩:2 のアイテムが発見できる．⟨pattern⟩:*count* はそれぞれペアになるパターンと支持度の値を表す．

Step 2: Distribute the search space.

*projected database(*射影データベース*)* を Step1 から得た 5 つの Prefix を元に，5 つのサブセット，Prefix ⟨PE⟩ を頭に持つセット，⋯，Prefix ⟨TS⟩ を頭に持つセットに分類する．

Step 3: Find subsets of sequential patterns.

構築された対応する 5 つの射影データベースに対し，再帰的に各々の手順を繰り返すことで，頻出する系列パターンを発見する．

Prefix ⟨PE⟩ を頭に持つセットを例として説明する．このとき，⟨PE⟩-projection データベースは，5 つの Postfix シーケンス ⟨WO TR⟩, ⟨TR WO⟩, ⟨TR TS WO⟩, ⟨PE TR WO BK⟩, ⟨WO TR WO⟩ を持つ．

まず，Step1 に戻り，⟨PE⟩-projection データベースをスキャンすることで，全ての ⟨PE⟩ を頭に持つ 2 系列パターンのシーケンスを発見する．すなわち，⟨PE WO⟩:5，⟨PE TR⟩:5 が発見される．

次に，これら 2 つの Prefix ⟨PE WO⟩，⟨PE TR⟩ に基づき，⟨PE⟩-projection データベースを 2 つのサブセットに分割する．

最後に，構築された各射影データベースを再帰的にスキャンする．ここで，Prefix ⟨PE WO⟩ は，3 つの Postfix のシーケンス ⟨TR⟩，⟨BK⟩，⟨TR WO⟩ を持つ．これらのシーケンスをさらにスキャンし，最終的に 3 系列パターン ⟨PE WO TR⟩:2 を発見する．このとき，最小支持度 *min_sup* 以下になるため，これ以上のスキャンは行わない．また，Prefix ⟨PE TR⟩ は，4 つの Postfix のシーケンス ⟨WO⟩, ⟨TS WO⟩, ⟨WO BK⟩，⟨WO⟩ を持ち，同様に 3 系列パターン ⟨PE TR WO⟩:4 を発見する．

このように，PrefixSpan アルゴリズムは再帰的に射影を行い，頻出するサブシーケンス，すなわち，系列パターンを発見できる．スキャンが終了し，最終的に出力される射影データベースと系列パターンを表 2.3 に示す．

表 2.3: Sequential patterns

Prefix	Projected Databases	Sequential Pattern
⟨PE⟩	⟨WO TR⟩, ⟨TR WO⟩	⟨PE⟩:5
	⟨TR TS WO⟩, ⟨PE TR WO BK⟩,	⟨PE TR⟩:5
	⟨WO TR WO⟩	⟨PE TR WO⟩:4
		⟨PE WO⟩:5
		⟨PE WO TR⟩:2
⟨WO⟩	⟨TR⟩, ⟨BK⟩	⟨WO⟩:5
		⟨WO TR⟩:2
⟨TR⟩	⟨WO⟩, ⟨TS WO⟩,	⟨TR⟩:5
	⟨WO BK⟩, ⟨WO⟩	⟨TR WO⟩:4
⟨BK⟩	⟨PE TR TS WO⟩	⟨BK⟩:2
⟨TS⟩	⟨WO⟩, ⟨PE PE TR WO BK⟩	⟨TS⟩:2
		⟨TS WO⟩:2

第3章
実験データ

3.1 CCC DATAset

CCC DATAset とは，客観的な評価と研究成果の共有を容易にすることを目的に，サイバークリーンセンターより提供されている研究用データセットである [10]．マルウェアの解析技術の研究のための「マルウェア検体」，感染手法の検知ならびに解析技術の研究のための「攻撃通信データ」，ボットの活動傾向把握の技術のための「攻撃元データ」の三つから構成される．

本論文では，CCC DATAset 2008～2011 攻撃通信データと攻撃元データを使用し，各データを照らし合わせる事で実験データとする．ハニーポットと各データの概要を以下に示し，各年度ごとのデータの差異を表 3.1 に示す．

表 3.1: CCC DATAset 2008，2009，2010，2011 の差異比較 [10]

攻撃通信データ				
項目	2008	2009	2010	2011
ハニーポット	honey001, 002	honey003, 004	honey001, 002	honey001, 002
収集日	2008/4/28 2008/4/29	2009/3/13 2009/3/14	2010/3/5 ～2010/3/5	2010/8/18～31 2011/1/18～31
総パケット数	15,901,943	3,511,850	22,486,674	23,009,309
攻撃元データ				
項目	2008	2009	2010	2011
ハニーポット数	112 台	94 台	92 台	72 台
ハニーポット ID	なし[3)]	あり	あり	あり
収集期間	2007/11/1 ～2008/4/30	2008/5/1 ～2009/4/30	2009/5/1 ～2010/4/30	2010/5/1 ～2011/1/31
全レコード数	2,942,221	2,470,766	1,162,093	158,734

[3)]ダウンロードホストと通信方向のみ．

3.1.1 ハニーポット

マルウェアを調査，研究するために設置するコンピュータである．コンピュータはマルウェアに侵入されやすいように脆弱に設定されており，侵入されてもマルウェアは外部に攻撃できない．また，攻撃者やマルウェアなどが介入できない方法で詳細な記録を取得する．

CCC DATAset では，円滑なデータ収集を行うため，感染の有無に関わらず，約 20 分間隔で定期的にシステムを再起動し，クリーンな状態にリセットして運用している．

3.1.2 マルウェア検体

ハニーポットで収集したマルウェア検体のハッシュ値（MD5，SHA1）をテキスト形式で記載したファイルである．

3.1.3 攻撃通信データ

攻撃通信データは，2 台のハニーポットを用いて観測したボットネットとの通信を tcpdump でパケットキャプチャーした libpcap 形式のファイルである．ハニーポットは 1 台のホスト OS 上で動作する Windows 2000 と XP の 2 台のゲスト OS により構成されている．それぞれインターネット接続されており，パケットキャプチャーはホスト OS 上で行われている．

3.1.4 攻撃元データ

攻撃元データは，112〜72 台のハニーポットで記録したマルウェア取得時のログで，csv 形式のファイルである．マルウェア検体の取得時刻，送信元 IP アドレス，送信元ポート番号，宛先 IP アドレス，宛先ポート番号，使用プロトコル（TCP もしくは UDP），マルウェア検体のハッシュ値（SHA1），マルウェア名称，ファイル名，以上の 11 項目を 1 レコードとして記録している．

3.2 実験データの作成

3.2.1 タイムスロット

実験にあたり，攻撃通信データを Windows XP が送信する NTP パケットを利用し，各データを約 20 分間隔で分割した [15]．これをタイムスロット（以下，スロット）と定義する．同様に，攻撃元データも分割する．ただし，攻撃元データには基準となる値がないため，単純に 20 分間隔で分割している．

3.2.2 トランザクションデータ

Apriori，PrefixSpan を適用するために，前処理として攻撃元データを加工し，トランザクションデータを作成した．トランザクションデータは，分割した各スロットを 1 つのトランザクションとし，その間にダウンロードされたマルウェアの種類をそのトランザクションに生じるアイテムとする．タイムスロットごとのマルウェアリストの一部を表 3.2 に示す．

本論文では，1 日分，すなわち，72 スロット分のトランザクションを纏めたデータを 1 つの入力データとし，頻出するアイテムの抽出を行う．

表 3.2: 攻撃通信データから抽出したトランザクションデータの例

スロット	マルウェア名
0	TROJ_SYSTEMHI.BQ
1	KDR_AGENT.ANHZ UNKNOWN TROJ_SYSTEMHI.BQ BKDR_AGENT.ANHZ UNKNOWN
2	PE_BOBAX.AH
3	PE_BOBAX.AH UNKNOWN BKDR_AGENT.ANHZ
⋮	
15323	PE_VIRUT.AV TROJ_IRCBRUTE.BW WORM_AUTORUN.CZU
15324	UNKNOWN PE_VIRUT.AV PE_VIRUT.AV WORM_AUTORUN.CZU TROJ_IRCBRUTE.BW

第4章
Apriori を用いた連携感染の抽出

4.1 概要

第 4 章では，大規模データからのマイニング技術であるアソシエーション分析を適用する事で，連携感染に関する価値ある相関ルールの抽出を試みる．アソシエーション分析（**相関ルール**）には，Agrawal らによって提案された Apriori アルゴリズムがあり，支持度（Support）と確信度（Confidence）の最小値を設定することで，膨大な組み合わせを効果的に枝狩りして効率よくルールを抽出できる技術として広く知られている．

本稿では，その実装として Christian Borgelt による Apriori Program[16] を活用して，攻撃元データに適用し，1 年間に頻出している共通の攻撃パターンを抽出する．攻撃パターンには，ダウンロードサーバ間の連携によるものと，マルウェアの種類の組み合わせによるものがあり，それぞれについて，観測装置（ハニーポット）による差や観測時期による差が生じるか検証する．

4.2 調査項目

第 4 章では実験データとして，CCC DATAset 2009 の攻撃通信データ，攻撃元データを使用する．調査は Honey003 (Windows XP+SP1) に限定しており，Honey00 4(Windows 2000) では行なっていない．

調査項目を以下に示す．

1. 攻撃通信データに対するマルウェアの相関ルール抽出
2. 攻撃通信データに対するダウンロードサーバの相関ルール抽出
3. 攻撃元データから抽出するマルウェアの相関ルールの観測地点間の差
4. 攻撃元データから抽出するマルウェアの相関ルールの観測時期の差

攻撃通信データは攻撃元データとマッチングを行う事により，マルウェア名を判定する事が可能である．そのため短期間ではあるが，2009 年 3 月 13 日，14 日に関しては正確なデータが得られ，精度の高い相関ルールが期待できる．本調査では，スロットごとにマルウェア名及びダウンロードサーバの IP アドレスを抽出し，2 つの観点からアソシエーション分析を行う事で，連携感染の関連性の強さを調査する．

一方，攻撃元データは 94 台のハニーポットを使用し，1 年間観測したデータである．このデータを使用する事により，抽出された相関ルールがどの位一般的であるか，すなわち，異なるハニーポットで共通に観測されるかどうか，あるいは，観測期間による差異は生じるかを明らかにする事ができる．すなわち，ハニーポット間及び長期間での 2 つの観点からアソシエーション分析を行う．

しかし，Honey003 及び 004 を除いて，正確にスロットに分割するのは困難である．誤差を許容して，単純に 20 分ごとにスロットを分割して分析を行う．

4.3 調査結果

4.3.1 攻撃通信データに対するマルウェアの相関ルール抽出

本節では，マルウェアの連携感染の関連性を調査する．表 3.1 をアソシエーション分析した結果を表 4.1 に示す．最小支持度 10 %以上，最小確信度 80 %以上の全ての相関ルールを示している[1)]．ここで，支持度は 145 個中その相関ルールが生じたスロットの割合，確信度は前件部のマルウェアを含むスロットのうち，結論部のマルウェアもダウンロードしているものの割合を表している．

表 1.1 の連携した攻撃パターン PE_VIRUT.AV ⇒ TROJ_BUZUS.AGB, WORM_SWTYMLAI.CD というルールは発見されなかったが，類似した相関ルール PE_VIRUT.AV , TROJ_BUZUS.AGB ⇒ WORM_SWTYMLAI.CD 及び PE_VIRUT.AV , WORM_SWTYMLAI.CD ⇒ TROJ_BUZUS.AGB は高い確信度で発見された．また，表 4.1 の結果から連携感染として，TROJ_BUZUS.AGB と WORM_SWTYMLAI.CD は特に強い関連性があると考えられる．

表 4.1: 攻撃通信データに対するマルウェアの相関ルール

No.	前件部			結論部	支持度	確信度
1		TROJ_BUZUS.AGB	⇒	WORM_SWTYMLAI.CD	41.4	100
2		WORM_SWTYMLAI.CD	⇒	TROJ_BUZUS.AGB	46.6	88.9
3	TROJ_BUZUS.AGB	BKDR_POEBOT.GN	⇒	WORM_SWTYMLAI.CD	10.3	100
4	WORM_SWTYMLAI.CD	BKDR_POEBOT.GN	⇒	TROJ_BUZUS.AGB	10.3	100
5	PE_VIRUT.AV	TROJ_BUZUS.AGB	⇒	WORM_SWTYMLAI.CD	29.3	100
6	PE_VIRUT.AV	WORM_SWTYMLAI.CD	⇒	TROJ_BUZUS.AGB	29.3	100

[1)]関連性が高いルールを抽出するため，支持度を低く，確信度を高く設定している．

4.3.2 攻撃通信データに対するダウンロードサーバの相関ルール

本節では，4.3.1 節と同様にダウンロードサーバの観点から関連性を調査する．ダウンロードサーバの IP アドレスで，アソシエーション分析を行った結果を表 4.2 に示す．最小支持度 10 %以上，最小確信度 50 %以上である[2)]．対応順位は，表 1.2 から割り出している．ここで，支持度は相関ルールのダウンロードサーバと通信していたスロットの割合を，確信度は前件部のダウンロードサーバからダウンロードしていたもののうち，結論部のダウンロードサーバも通信しているスロットの割合を表している．

分析の入力データは，表 3.1 のマルウェア名を対応する IP アドレスに置き換えたものだが，マルウェア名とダウンロードサーバは 1 対 1 で対応している訳ではない．例えば，PE_VIRUT.AV は 16 種類の IP アドレスからダウンロードされている．

No.1，2 の IP アドレスは PE_VIRUT.AV のものだが，114.145.**.166 は 12 回，122.18.***.123 は 21 回使用されており，16 種類のうち上位の 2 種類である．また，No.3，4 から IP アドレスの観点から見た場合でも連携感染として，TROJ_BUZUS.AGB と WORM_SWTYMLAI.CD の間には強い関連性がある事が分かる．しかし，複数のダウンロードサーバを使用するマルウェア，すなわち，PE_VIRUT.AV を含む表 1.1 に関するルールは抽出できなかった．

表 4.2: 攻撃通信データに対するダウンロードサーバの相関ルール

No.	前件部		結論部	支持度	確信度	対応マルウェア	対応順位
1	114.145.**.166	⇒	122.18.***.123	12.1	85.7	PE ⇒ PE	
2	122.18.***.123	⇒	114.145.**.166	15.5	66.7	PE ⇒ PE	
3	67.215.*.206	⇒	72.10.***.195	46.6	100	TROJ ⇒ WORM	TOP5 ⇒ TOP2
4	72.10.***.195	⇒	67.215.*.206	46.6	100	WORM ⇒ TROJ	TOP2 ⇒ TOP5

[2)]確信度を 80 %から 50 %に下げたのは，抽出されたルールが少なかったためである．

4.3.3 攻撃元データから抽出するマルウェアの相関ルールの観測地点間の差

本節では，4.3.1 節の相関ルールが異なるハニーポットで共通に観測されるかを調査する．全 94 のハニーポット ID でアソシエーション分析を行った結果を表 4.3 に示す．2009 年 3 月 13 日のみのデータで分析を行っており，閾値はスロット数 3 以上，確信度 80 %以上である[3)]．例えば，No.1 のルールのハニーポット数 32 は，全 94 台のハニーポットのうち，このルールを 1 回でも抽出したハニーポットが 32 台あることを表している．このとき，支持度と確信度の違いは無視している．

上位の相関ルールの出現数（相関ルールが成立するスロットの数）を図 4.1 に示す．X 軸はハニーポット数を k とし，Y 軸は k 以上のハニーポットで観測された異なる相関ルールの個数 $N(k)$ を表す．

結果は，TROJ_BUZUS.AGB と WORM_SWTYMLAI.CD の相関ルールが上位に抽出されたことを示しており，概ね 1 台のハニーポットで観測した結果と矛盾がなかった．3 分の 1 以上のハニーポットにて，TROJ_BUZUS.AGB と WORM_SWTYMLAI.CD 間のルールが得られている．

表 4.3: 2009 年 3 月 13 日に観測されたマルウェアに関する相関ルールのハニーポット数

No.	前件部			結論部	ハニーポット数
1		TROJ_BUZUS.AGB	⇒	WORM_SWTYMLAI.CD	36
2		WORM_SWTYMLAI.CD	⇒	TROJ_BUZUS.AGB	36
3	TROJ_BUZUS.AGB	BKDR_VANBOT.AHH	⇒	WORM_SWTYMLAI.CD	12
4	WORM_SWTYMLAI.CD	BKDR_VANBOT.AHH	⇒	TROJ_BUZUS.AGB	12
5		TROJ_DLOADR.CBK	⇒	UNKNOWN	8
6	TROJ_BUZUS.AGB	PE_VIRUT.AV	⇒	WORM_SWTYMLAI.CD	7
7	WORM_SWTYMLAI.CD	PE_VIRUT.AV	⇒	TROJ_BUZUS.AGB	7
8	PE_VIRUT.AV	TROJ_BUZUS.AGB	⇒	WORM_SWTYMLAI.CD	6
9		TROJ_AGENT.ANDF	⇒	UNKNOWN	6
10	PE_VIRUT.AV	WORM_SWTYMLAI.CD	⇒	TROJ_BUZUS.AGB	6

[3)]支持度の代わりにスロット数にしたのは，スロットごとに抽出されるルール数が違うため，ルール総数で正規化する支持度で揃えては共通のルールを抽出するのに不都合だったからである．

また，図 4.1 から，共通したルールばかりでなく，多様なルールが数多く生じることが分かる．広範囲で観測されたルールは特に連携感染を行っている可能性が高いと考えられ，これらルールにはマルウェアに偏りがみられる事から，特定のマルウェアのみが連携感染を行っていると考える．

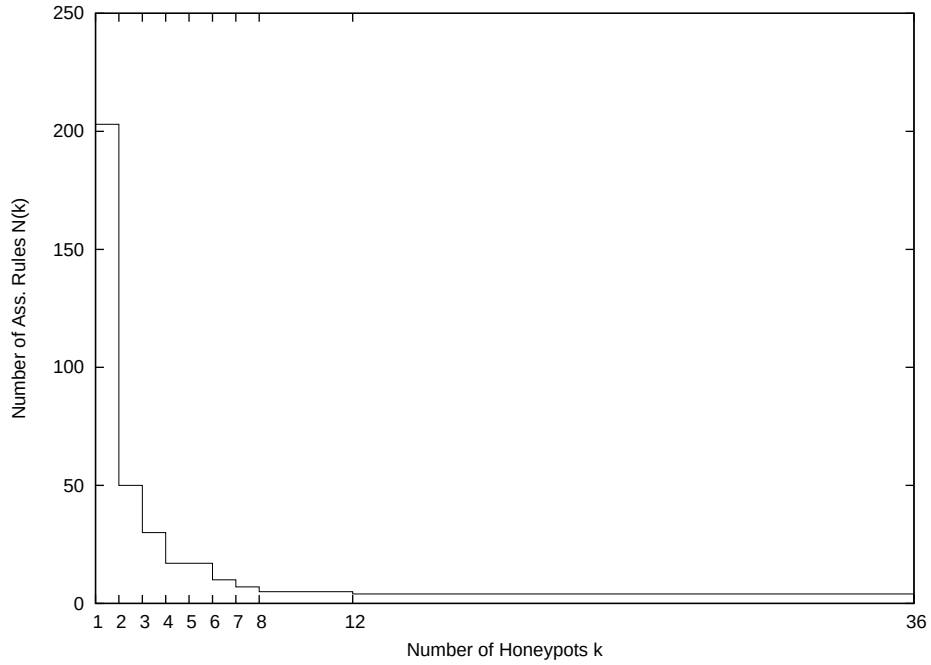


図 4.1: 上位 10 位の相関ルールの出現スロット数

4.3.4 攻撃元データから抽出するマルウェアの相関ルールの観測時期の差

本節では，観測期間による相関ルールに差異が生じるかを調査する．1 年間の Honey003 でアソシエーション分析を行った結果を表 4.4 に示す．閾値はスロット数 3 以上，確信度 80 %以上で 365 日全ての相関ルールを抽出し，該当したマルウェア名を含むルール数を月ごとに纏めたものである．4.3.3 節と同様に支持度と確信度の違いは無視している．また，抽出したマルウェア名はこれまでの調査でが強いと考えられる PE_VIRUT.AV（PE），TROJ_BUZUS.AGB（TROJ），WORM_SWTYMLAI.CD（WORM）とした．

1 年間で観測された PE，TROJ，WORM のいずれかを含む以下の上位 3 個の相関ルール数の推移を図 4.2 に示す．

1. PE_VIRUT.AV WORM_SWTYMLAI.CD ⇒ TSPY_KOLABC.CH
2. TROJ_BUZUS.AGB ⇒ WORM_SWTYMLAI.CD
3. TSPY_KOLABC.CH ⇒ WORM_SWTYMLAI.CD

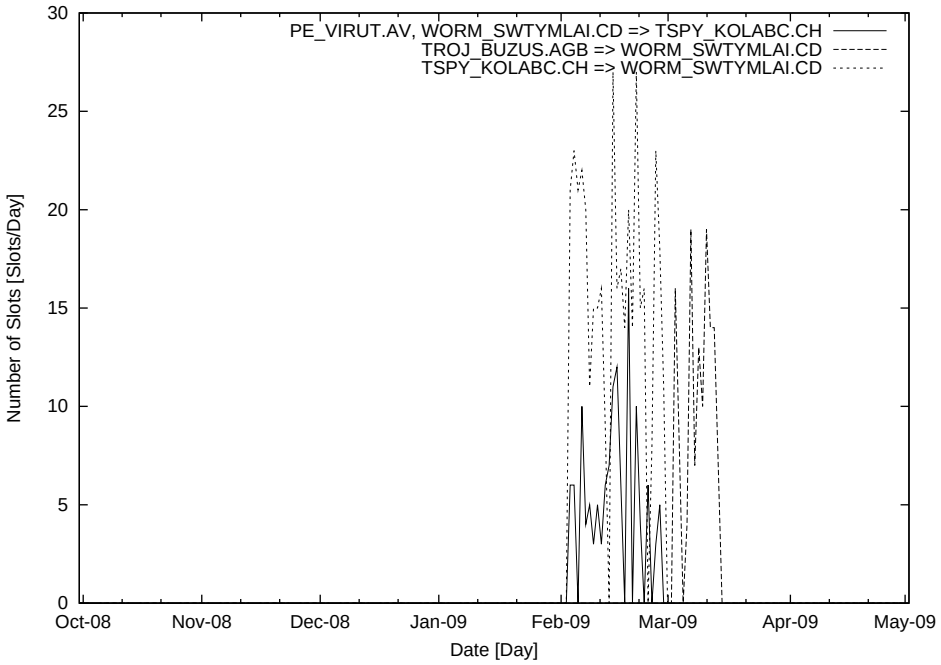


図 4.2: PE，TROJ，WORM のいずれかを含む上位 3 個の相関ルール数の推移

表 4.4: 1 日ごとに観測された PE，TROJ，WORM を含むルール数

月	PE	TROJ	WORM
2008/05	31	0	0
2008/06	76	0	0
2008/07	111	0	0
2008/08	5	0	0
2008/09	8	0	0
2008/10	44	0	0
2008/11	27	0	0
2008/12	35	0	0
2009/01	135	0	0
2009/02	125	0	226
2009/03	79	53	74
2009/04	30	0	0

表 4.4 より，PE_VIRUT.AV は年間を通して観測されており，関係する多くのルールが抽出できている．その中でも 1 のルールより，WORM_SWTYMLAI.CD と関係するルールが一番多かった事から，長期間でも連携感染として，強い関連性があると考えられる．また，2，3 のルールより，TSPY_KOLABC.CH とのルールが上位に来ており，4.3.1 節では見られなかった関連性，すなわち，観測期間による差異が伺える．

1 年間で観測された UNKNOWN を含まない相関ルールのうち，以下の上位 3 個のルールの推移を図 4.3 に示す.

1. BKDR_VANBOT.HI ⇒ BKDR_SDBOT.BU
2. BKDR_POEBOT.AHP ⇒ TROJ_QHOST.WT
3. TSPY_KOLABC.CH ⇒ WORM_SWTYMLAI.CD

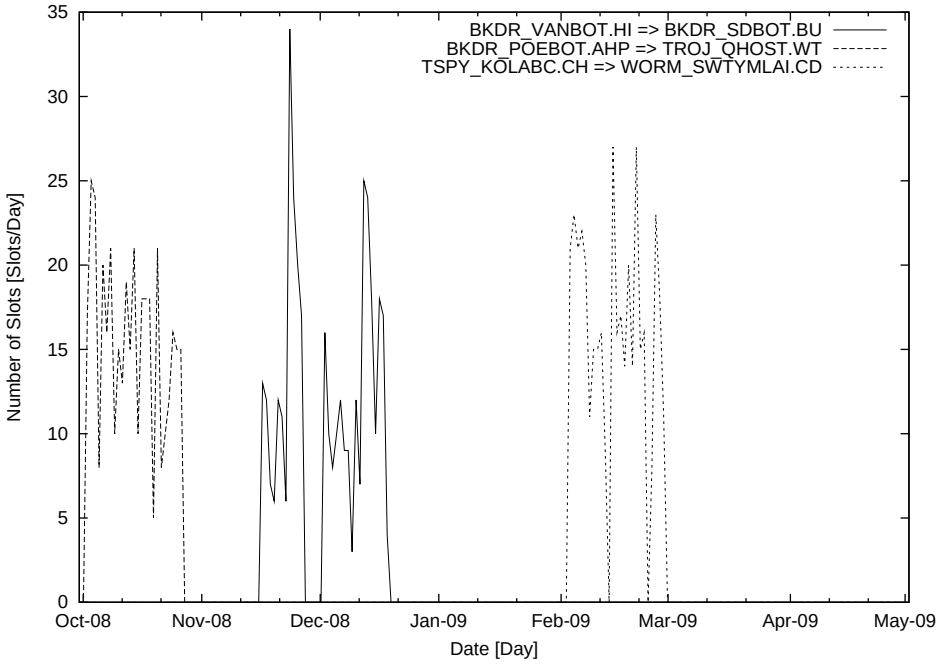


図 4.3: 1 年間の UNKNOWN を含まない上位 3 個の相関ルール数の推移

これは，年間で最も観測されたルールであるが，ルールの出現期間は短く，1 年間を通して観測されてはいない．この事から，連携感染期間は短い事が分かる．理由としては，新たなマルウェアが出現したり，更新されたりするために，特定のマルウェア間の連携期間が短くなってしまう事が考えられる．

4.4 まとめ

連携してマルウェアをダウンロードしているサーバやマルウェアに関する規則を，機械的に抽出する方式を提案した．また，通信データを詳細に分析して得られる結果とほぼ同じ結果が抽出できる事を実証した．共通に現われたのは，PE_VIRUT.AV, TROJ_BUZUS.AGB, WORM_SWTYMLAI.CD 間の強い関係である．

本実験を総合して，TROJ と WORM の関連性が強い相関ルールが最も頻出していた．広範囲で観測されたルールは，連携感染を行っている可能性が高い事を示し，長期間で観測した結果では，連携感染期間は短い事がわかった．ダウンロードサーバ間の相関ルールを抽出する事は，複数のダウンロードサーバを使用する PE_VIRUT.AV 等があるため難しいが，4.3.1 節と 4.3.2 節の結果から，マルウェアの相関ルールを抽出し，それを IP アドレスに適用する事が有効であると考えられる．

Honey003 及び 004 を正確に分割した場合と単純に分割した場合で比較した結果，支持度及び確信度の違いはあったが，抽出された上位の相関ルールは同じだったため，分析結果は妥当であると考える．

第5章
PrefixSpan を用いた連携感染の抽出

5.1 概要

第 4 章で，アソシエーション分析を適用し，関連性の強いマルウェアの組み合わせ，すなわち，連携感染を自動検出する手法を提案した．しかし，Apriori による結果は，あくまでアイテムの集合であり，順序は考慮していない．例えば，PE_VIRUT.AV, TROJ_BUZUS.AGB, WORM_SWTYMLAI.CD という 3 種類のマルウェアがどの順序で感染するのかはわからない．TROJ_BUZUS.AGB, WORM_SWTYMLAI.CD, PE_VIRUT.AV という順序や WORM_SWTYMLAI.CD, PE_VIRUT.AV, TROJ_BUZUS.AGB という順序で感染する可能性も考えられる．

また，ボットネットによる攻撃を早期に検出するには，マルウェアがどの順序で感染しているのかを知ることが重要である．実際には，4 章で示した通り，PE_VIRUT.AV が最初に感染するマルウェアだが，このように感染の順序がわかれば，PE_VIRUT.AV を検出した時点で，ボットネットによる攻撃と判定できる．

そこで，第 5 章では，検出したパターンを構成するマルウェアのアイテム数，シーケンスの長さに着目し，PrefixSpan アルゴリズムを使用して時系列を考慮したパターンの抽出を試みる．なお，時系列を考慮するにあたり，第 5 章では連携感染と明確に区別するため，連携感染のことを**系列感染パターン**と呼ぶ．

5.2 系列感染パターン

5.2.1 最小支持度の設定

パターン数とパターンの長さの関係を図 5.1 に示す．縦棒は最小支持度によってパターン数が変わることを表す．図 5.1 より，パターンの長さは 2〜3 にかけて多い傾向だった．この結果はデータマイニングを行う際，支持度を調整するのに有効である．すなわち，シーケンスの長さを設定するための基準として役立つ．検出される系列感染パターンの数は，最小支持度によって決まるためである．

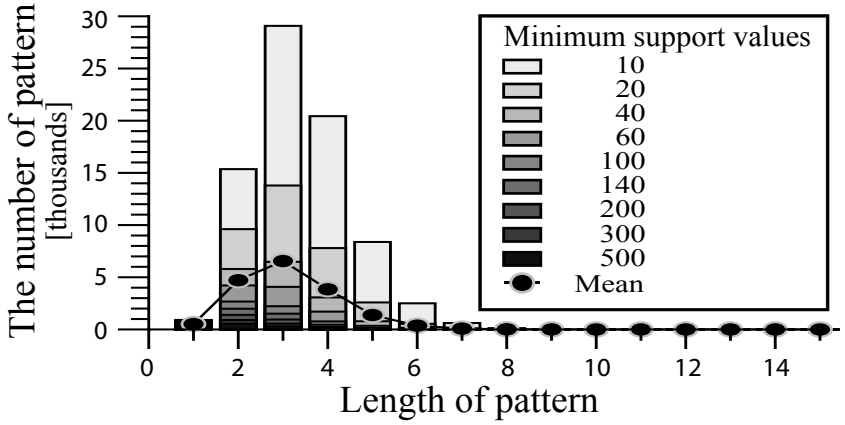


図 5.1: 系列感染パターンの数とパターンの長さの関係

また，系列感染パターンを抽出するにあたって，*PrefixSpan* アルゴリズムを CCC DATAset に適用し，マルウェアの系列感染パターンのリストを作成する．このとき，リストはマルウェアに感染しているスロット数に従ってソートする．ハニーポットが 20 分ごとに再起動するので，1 日分で 72 スロット分のデータとなることを考慮して，系列感染パターン (2*length*) の最小支持度を設定する必要がある．例えば，ある日に集中的な攻撃があった場合，感染スロット数は 72 以下か等しい．そのため，系列感染パターン (2*length*) を発見するための最小支持度は，70 が妥当であると考える．その上で，図 5.1 に示すシーケンスの長さの分布の傾向を参照すると，系列感染パターン (3*length*) の最小支持度は，系列感染パターン (2*length*) の最小支持度の約 40 %程度，すなわち，30 が良いと考えられる．

5.2.2 系列感染パターンの定義

発見した系列感染パターンに $P_{x.y}$ というインデックスを定義する．x はシーケンスの長さ，y は全てのパターンをソート後，番号を与えたリストの番号である．例えば，パターン $P_{3.1203}$ は，シーケンスの長さが 3，1203 番目の系列感染パターンであることを表す[1)]．

なお，系列感染パターンのインデックスを定義するためにはパターンを網羅しなければならない．図 5.1 に示す通り，最小支持度によって発見されるパターン数が決まるので，パターンを調査するために最小支持度を 50 %に減らした．すなわち，2*length* が 35，3*length* が 15 とする．

また，PrefixSpan で系列感染パターンを抽出した結果から，**重複**と**非重複**の 2 つのカテゴリに分類できることがわかった．重複は非重複と異なり，2 つ以上の重複するマルウェアが存在することを表す．例えば，図 5.2 (a) のパターン $P_{2.386}$，$P_{2.300}$，図 5.3 (a) のパターン $P_{3.1203}$，$P_{3.857}$ は, それぞれ `PE_VIRUT.AV` と `PE_BOBAX.AK` が重複している．

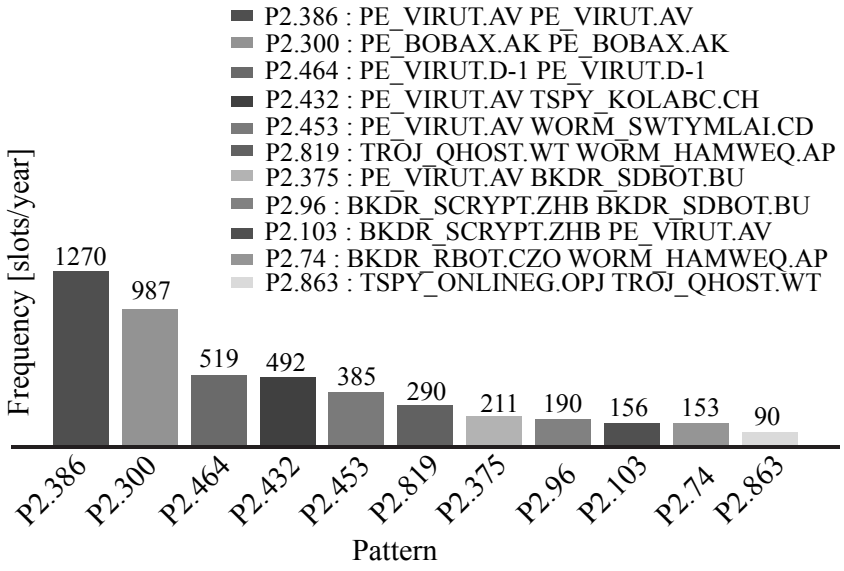
5.3 調査項目

第 5 章でも実験データとして，同様に CCC DATAset 2009 の攻撃通信データ，攻撃元データを使用する．図 5.2，図 5.3 より，系列感染パターンは全てのハニーポットで同様の振る舞いを行なっている可能性が高い．したがって，これら 2 つの振る舞いが一般的な系列感染パターンを表すのに十分であると考える．そこで，本調査では 94 台のハニーポットのうち，Honey003 (Windows XP+SP1) と Honey004 (Windows 2000) の 2 つのハニーポットに焦点をあて，調査を行う．

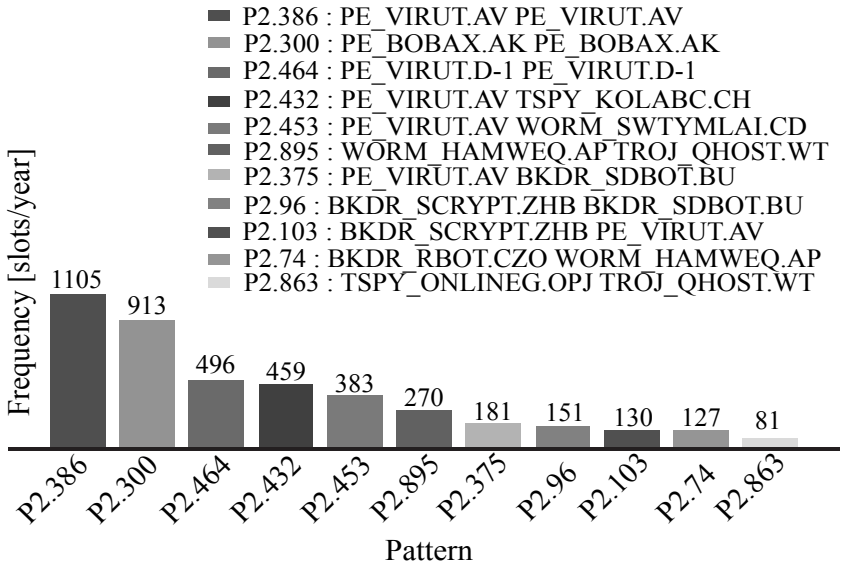
調査項目を以下に示す.

1. ハニーポット間における系列感染パターン
2. IP アドレスとダウンロード時間に基づく系列感染パターン
3. 1 年間の系列感染パターンの活動分布
4. PrefixSpan のパフォーマンス
5. 系列感染パターンのエントロピー解析

[1)] 1203 番目に頻出したパターンではない点に注意．番号はあくまでパターンを識別するためのものである．

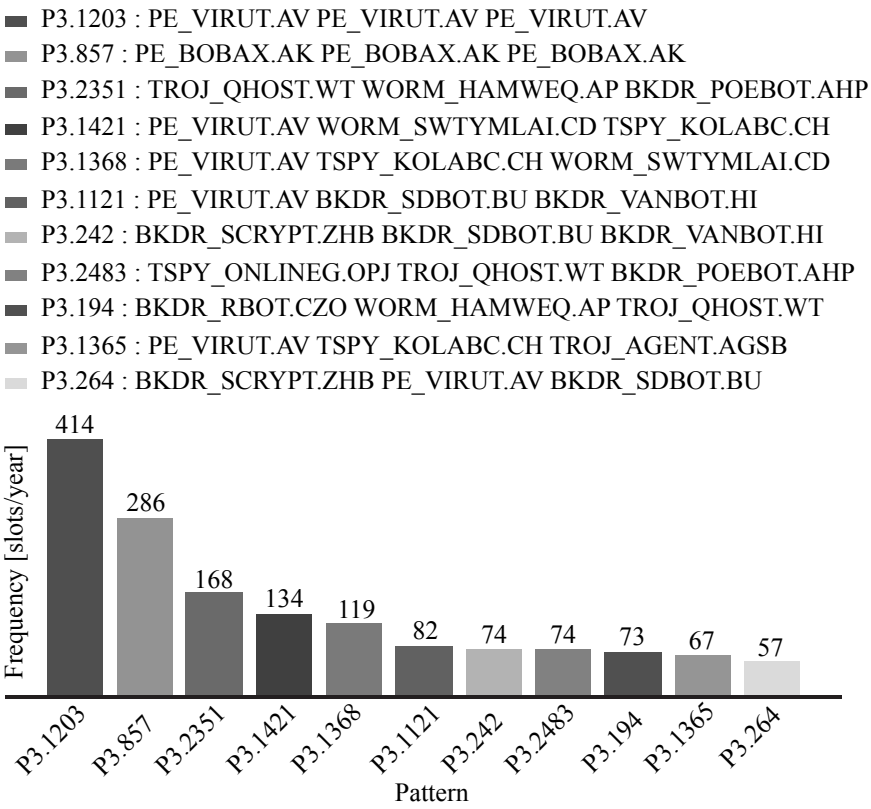


(a) Honey003 (XP)

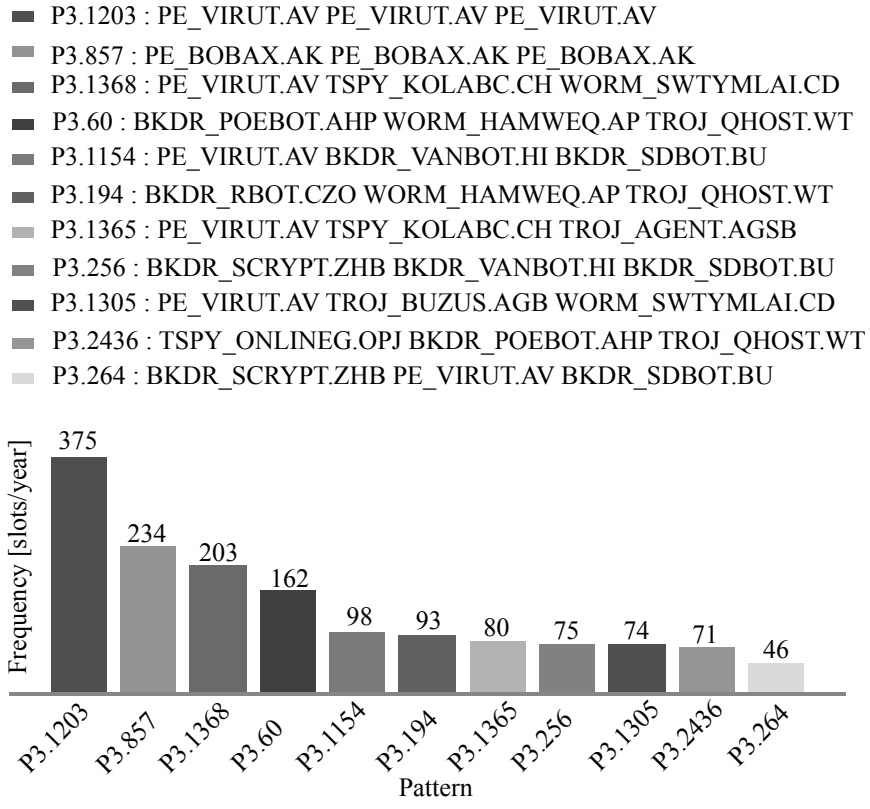


(b) Honey004 (2000)

図 5.2: 94 台のハニーポットで観測した系列感染パターン (2*length*)



(a) Honey003 (XP)



(b) Honey004 (2000)

図 5.3: 94 台のハニーポットで観測した系列感染パターン (3*length*)

5.4 調査結果

5.4.1 ハニーポット間における系列感染パターン

図 5.2 (a)，(b) は，2 つのハニーポットにおける系列感染パターン (2*length*) である．それぞれ Honey003，004 であり，X 軸はパターン名，Y 軸はダウンロード数の頻度を表す．図 5.2 から，系列感染パターンの順位，ダウンロード数の頻度という点でこれら 2 つのハニーポットポット間に有意な関係があることがわかる．

また，図 5.2 にて，上位 5 位までの感染パターンが両方のハニーポットで共通して出現している．それ以下のパターンも僅かなスロット数の違いしか見受けられず，各系列感染パターンのダウンロード数の頻度による差も比較的小さい．例えば，パターン $P_{2.453}$ のダウンロード数の頻度は，Honey003 と 004 でそれぞれ 385 と 383 スロットしかなく，2 スロットの違いしかなかった．

図 5.3 (a)，(b) は，系列感染パターン (3*length*) の結果を示している．それぞれ 169，118 のパターン (29 %と 26 %の非重複パターンを含む) が抽出されていた．

また，2*length*，3*length* の両方にてランキング上位のパターンは，`PE_VIRUT.AV` と `PE_BOBAX.AK` を含む重複するパターンであった．この重複は，マルウェアが単一のスロットに対して，複数回に渡ってハニーポットを感染させたことを意味する．以上の結果は，これらマルウェアがボットネットのシステムが採用する最も一般的なマルウェアであることを示唆している．

5.4.2 IP アドレスとダウンロード時間に基づく系列感染パターン

ボットネットはインターネットを通じてボットを配布する．その送信元 IP アドレスとダウンロード時間を使用し，マルウェアの拡散する挙動を学習することで，ボットネットの攻撃による脅威への警告が行えると考える．そこで，本項ではボットネットによって使用された送信元 IP アドレスとマルウェアのダウンロード時間を調査する．

まず，送信元 IP アドレスとダウンロード時間という特徴に基づき，系列感染パターン (3*length*) をいくつかのパターンに分類した．送信元 IP アドレスに基づく系列感染パターンの種類を表 5.1，マルウェアのダウンロード時間に基づく系列感染パターンの種類を表 5.2 に示す．*IP* **パターンコード**は感染したマルウェアの送信元 IP アドレスの順序の種類，**タイムパターンコード**はマルウェアをダウンロードした時系列の種類をそれぞれ表す．

表 5.1: 送信元 IP アドレスに基づく IP パターンコード

コード名	パターン		
A_1	$S1$	$S1$	$S1$
A_2	$S1$	$S1$	$S2$
A_3	$S1$	$S2$	$S1$
A_4	$S1$	$S2$	$S2$
A_5	$S1$	$S2$	$S3$

表 5.2: マルウェアのダウンロード時間に基づくタイムパターンコード

コード名	パターン		
E_1	$T1$	$T1$	$T1$
E_2	$T1$	$T1$	$T2$
E_3	$T1$	$T2$	$T2$
E_4	$T1$	$T2$	$T3$

次に，系列感染パターン (3*length*) の IP パターンの種類とユニークホスト数との関係を纏めた結果を表 5.3 に示す．IP パターンの種類は表 5.1，表 5.2 を元に表記する．例えば，$P_{3.242}$ というパターンは，タイプ A_3E_3 である．これは，1 番目と 3 番目のマルウェアが同じ送信元 IP アドレス (A_3) からダウンロードされ，2 番目と 3 番目のマルウェアが同時刻 (E_3) でダウンロードされていることを表す．タイプは主なタイプを記しており，少数のみ確認されたタイプは除外している．また，ユニークホスト数は送信元 IP アドレスを元にした．送信元 IP アドレスは，マルウェアを配布するダウンロードサーバと考えることができるためであ

る．一番左から 1 番目，続いて 2 番目，3 番目という順番で感染したホストを表す．さらに，ランキング上位のパターンをダウンロード時間の類似性を元に，2 つのグループに分類している．それぞれグループ A と B と定義する．

表 5.3: マルウェアの系列感染パターン (3*length*)

Honey	ID	系列感染パターン
003	$P_{3.2351}$	TROJ_QHOST.WT WORM_HAMWEQ.AP BKDR_POEBOT.AHP
	$P_{3.2483}$	TSPY_ONLINEG.OPJ TROJ_QHOST.WT BKDR_POEBOT.AHP
	$P_{3.194}$	BKDR_RBOT.CZO WORM_HAMWEQ.AP TROJ_QHOST.WT
004	$P_{3.60}$	BKDR_POEBOT.AHP WORM_HAMWEQ.AP TROJ_QHOST.WT
	$P_{3.2436}$	TSPY_ONLINEG.OPJ BKDR_POEBOT.AHP TROJ_QHOST.WT
	$P_{3.194}$	BKDR_RBOT.CZO WORM_HAMWEQ.AP TROJ_QHOST.WT
003	$P_{3.1121}$	PE_VIRUT.AV BKDR_SDBOT.BU BKDR_VANBOT.HI
	$P_{3.242}$	BKDR_SCRYPT.ZHB BKDR_SDBOT.BU BKDR_VANBOT.HI
	$P_{3.264}$	BKDR_SCRYPT.ZHB PE_VIRUT.AV BKDR_SDBOT.BU
004	$P_{3.1154}$	PE_VIRUT.AV BKDR_VANBOT.HI BKDR_SDBOT.BU
	$P_{3.256}$	BKDR_SCRYPT.ZHB BKDR_VANBOT.HI BKDR_SDBOT.BU
	$P_{3.264}$	BKDR_SCRYPT.ZHB PE_VIRUT.AV BKDR_SDBOT.BU

表 5.3 より，マルウェアはユニークな IP アドレス，あるいは複数の IP アドレスから送信されている．3*length* パターンのいくつかは 1 つの送信元 IP アドレスによるものだったが，2 つ以上の IP アドレス，すなわち，合計 3 台のダウンロードサーバに分散した攻撃も存在した．また，攻撃者のグループ A と B は，その送信元 IP アドレスの種類が異なっている．

例えば，グループ A はコード A_1，A_4 と E_1 を使用したものが多い．パターン $P_{3.2351}$ や $P_{3.194}$，$P_{3.60}$ の攻撃は概ね 1 つのユニークホストから，ほぼ同時刻にダウンロードされている．また，グループ A 内における傾向の違いとして，パターンを構成するマルウェアに特徴があった．パターン $P_{3.2483}$ と $P_{3.2436}$ を構成する TSPY_ONLINEG.OPJ である．これは，グループ A の他のパターンには存在しないマルウェアであり，1 番目に感染している．ダウンロードも，Honey003 は 41，2 は 31 という多くのユニークホストからされており，時間に関しても，2 番目と 3 番目のマルウェアとは異なる時間にダウンロードされている．

それに対して，グループ B の 3*length* パターンによる攻撃は，コード A_3，A_5 と E_3 が概ね一致している．これらのパターン $P_{3.1121}$，$P_{3.242}$，$P_{3.1154}$，$P_{3.256}$ は，2 つ以上の複数のホストからマルウェアをダウンロードする点が特徴的である．2 番目，3 番目のユニークホスト数は少ないが，1 番目のマルウェアは多くのユニークホストからダウンロードされている．また，グループ B では，2 台のハニーポットで共通してパターン $P_{3.264}$ を観測した．このパターンの大きな違いは，2 番目のユニークホスト数が多い点にある．パターン $P_{3.264}$ の 1 番目及び

Honey	ID	頻度	平均 [s]	標準偏差 [s]	ユニークホスト			タイプ	グループ
003	$P_{3.2351}$	168	4.27	51.07	1	1	1	A_1E_1	A
	$P_{3.2483}$	74	97.04	165.46	41	1	1	$A_4E_{1,3}$	A
	$P_{3.194}$	73	56.65	235.71	3	1	1	A_1E_1	A
004	$P_{3.60}$	162	34.12	175.92	8	1	1	A_1E_1	A
	$P_{3.2436}$	72.66	191.33	34	1	1	A_4E_3	A	
	$P_{3.194}$	71	381.48	478.60	5	1	1	$A_1E_{1,3}$	A
003	$P_{3.1121}$	82	108.31	212.90	48	1	1	$A_3E_{1,3}$	B
	$P_{3.242}$	74	732.12	422.57	11	1	1	$A_{3,5}E_3$	B
	$P_{3.264}$	57	862.60	304.87	5	42	1	$A_5E_{3,4}$	B
004	$P_{3.1154}$	98	75.54	177.64	55	1	1	A_5E_3	B
	$P_{3.256}$	75	821.86	326.30	6	2	1	$A_{2,5}E_3$	B
	$P_{3.264}$	46	968.42	258.12	6	34	1	$A_5E_{3,4}$	B

3 番目のマルウェアは，わずかなユニークホスト数だが，2 番目のマルウェア `PE_VIRUT.AV` は多くのユニークホスト数である．これはボットネットが攻撃を行うために，連携している 1 つの証拠と言える．

最後に，系列感染パターン 3.2483，$P_{3.264}$ による連携感染のタイムチャートを図 5.4 に示す．図 5.4 から，表 5.1，表 5.2 による送信元 IP アドレス，時系列が観測でき，連携感染が挙動どうなっているかがわかる．このように，系列感染パターンを定義し，動作をマッピングすることで，早期に脅威の識別及び予測を行うことができる可能性がある．例えば，図 5.4 の **MALWARE1** を検知することで，ネットワークを介したマルウェアの拡散を通知できる．

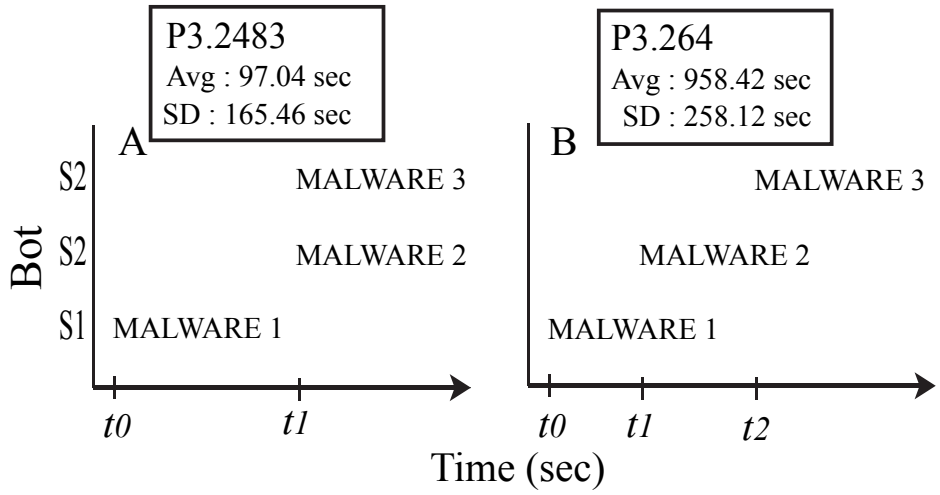
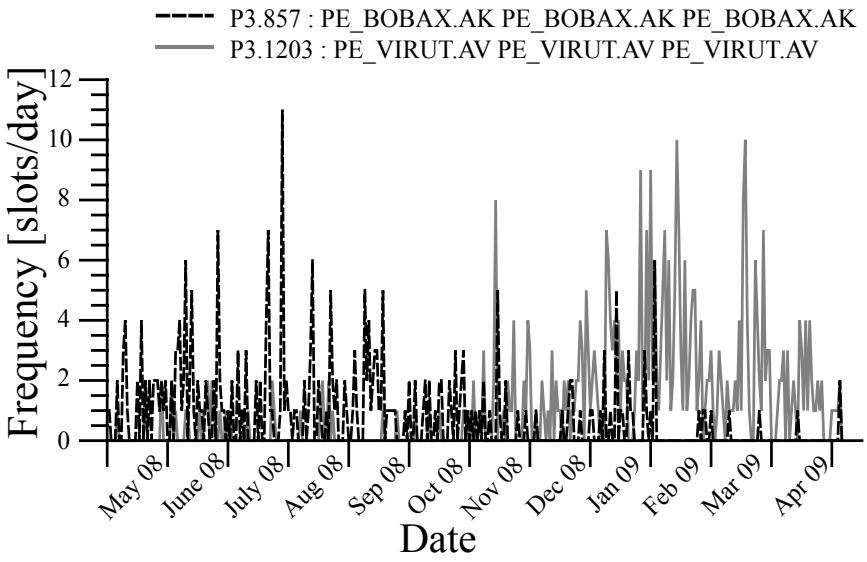


図 5.4: 系列感染パターン (3*length*) によって行われた連携感染のタイムチャート: (A) IP パターンコード A_4，タイムパターンコード E_3 に属するパターン $P_{3.2483}$，(B) IP パターンコード A_5，タイムパターンコード E_4 に属するパターン $P_{3.264}$

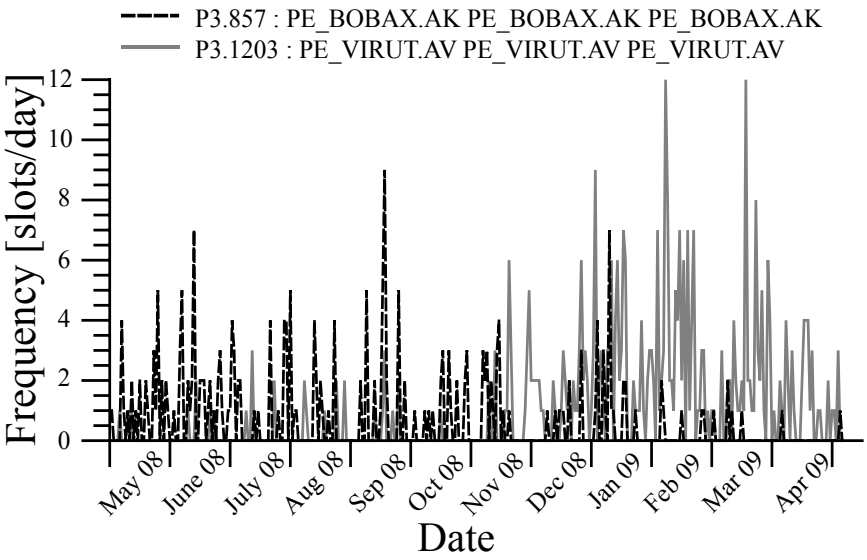
5.4.3 1 年間の系列感染パターンの活動分布

両方のハニーポットで共通して最も頻繁なダウンロードを観測した重複する系列感染パターン (3*length*) を図 5.5 に示す．X 軸は日付，Y 軸はダウンロード数の頻度を表す．最もよく見られた重複パターンは，`PE_VIRUT.AV` と `PE_BOBAX.AK` である．図 5.5 (a) より，パターン $P_{3.1203}$ は 2009 年 2 月，3 月に，10 スロット/日という高頻度で観測され，パターン $P_{3.857}$ は 2008 年 7 月に，11 スロット/日という最大の観測がされていることがわかる．同様に，図 5.5 (b) でも $P_{3.1203}$ の傾向は，12 スロット/日と頻度は高い．また，Honey004 のパターン $P_{3.857}$ は，2008 年 9 月に 9 スロット/日と高頻度である．

これら 2 つのパターン $P_{3.1203}$，$P_{3.857}$ は，WindowsXP，2000 を実行しているシステムのいくつかの機能，インターネット接続ファイアーウォールやインターネット接続の共有などを無効にするマルウェアが関連している．このマルウェアは様々なポートを開き，IRC サーバに接続する機能を持ち，被害及び感染力が中程度と評価されている [17]．したがって，図 5.5 はボットネットの攻撃に関するボットネットシステムの C&C サーバ活動を示すと考える．



(a) Honey003 (XP)



(b) Honey004 (2000)

図 5.5: 1 年間の重複する系列感染パターン (3*length*) の分布

5.4.4 系列感染パターンの分類

非重複の系列感染パターン (3*length*) は，マルウェア名かダウンロード時間のいずれかに似た傾向がある．

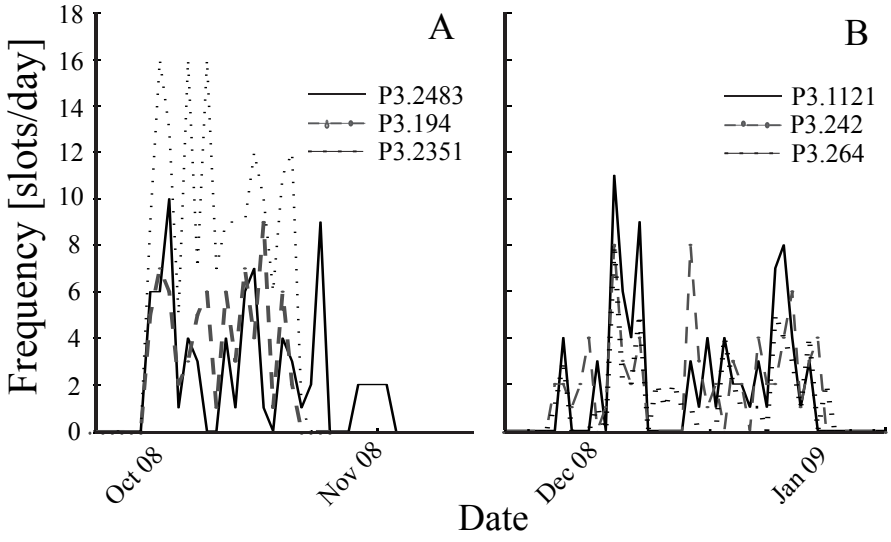
表 5.3 より，グループ A のパターンは，2 台のハニーポットで同じ 5 種類のマルウェアで構成されているが，マルウェアのパターンの感染順でいくつかの違いが見られる．また，いずれのハニーポットでも確認できるパターン $P_{3.194}$ は，ダウンロード数の頻度が 2 スロット異なる．同様にグループ B では，2 つのハニーポットで 4 種類のマルウェアだが，それらの感染順が部分的に逆転していることがわかる．ハニーポットで共通するパターン $P_{3.264}$ は，ダウンロード数の頻度に 11 スロットの違いがある．

グループ A と B の 1 年間における非重複の系列感染パターン (3*length*) の分布を図 5.6 に示す．両方のハニーポットでグループ A を観測し，2008 年 10 月に 20 日間の期間内でダウンロードされたことがわかった．グループ A は共通して，2008 年 10 月に約 20 日間に渡って攻撃されていることがわかる．図 5.6 (a-A)，図 5.6 (b-A) より，各ハニーポットにおける最大の感染はそれぞれ 16 スロット，22 スロットである．

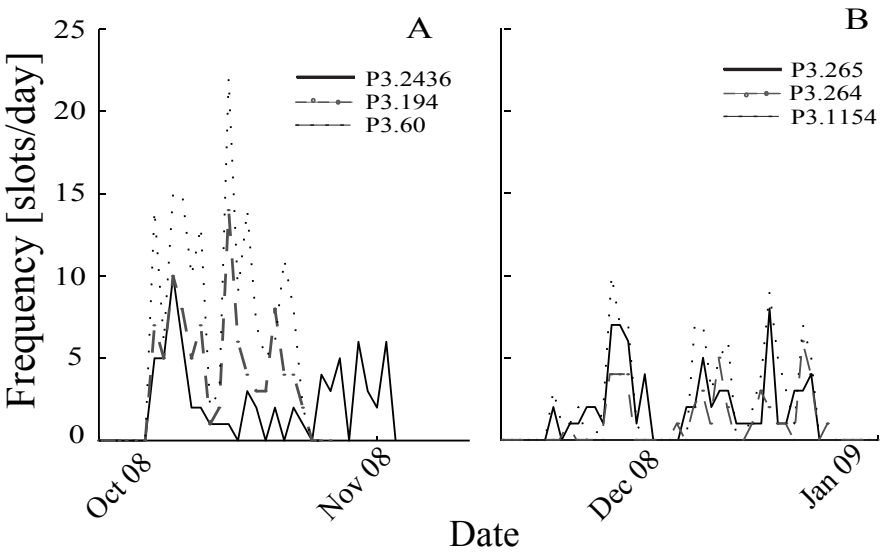
グループ B のボットネットによる攻撃活動は約 25 日間行われており，持続的な集中攻撃 [18] に似た傾向である．活動は 2008 年 12 月から 2009 年 1 月にかけて行われ，図 5.6 (a-B), 5.6 (b-B) で示す通り，最大の感染は Honey003 で 11 スロットだった．

調査結果より，2 つのハニーポット間に大きな類似性を発見した．重複しない系列感染パターンにおける 2 つの共通点を以下に示す．

1. 攻撃は 1 年間で 1 ヶ月未満という短期間に集中して行われる．
2. 感染スロット数は，重複する系列感染パターン (3*length*) より大きい．



(a) Honey003 (XP)



(b) Honey004 (2000)

図 5.6: 1 年間の非重複の系列感染パターン (3*length*) の分布

次に，系列感染パターン (3*length*) の感染時間間隔を調査する．感染時間間隔は系列感染パターンにおいて，最初から最後のマルウェアがダウンロードされるまでの時間の差と定義する．

図 5.5 に示した感染時間間隔の平均と標準偏差を参照してほしい．それぞれの平均感染時間間隔の分布は大きく異なっていた．これは，複数のダウンロードイベントや複数のボットネット攻撃によってネットワークトラフィックの動的挙動が変動し，感染時間間隔に差が起きたためだと考える．

系列感染パターンの感染時間間隔の分布を図 5.7 に示す．2 台のハニーポットで観測されたパターン $P_{3.194}$ は，感染時間間隔の平均が 7 分未満，標準偏差が 7 分以上だったが，図 5.7 より，非常に小さい値が大多数を占めていた．すなわち，感染時間間隔は非常に短期間である．この結果は，これらのパターンが継続的に一定の時間間隔で実行されていることを意味し，グループ A のパターンが同じボットネットシステムから送信されたことを示す．逆に，$P_{3.264}$ は，14 分以上の平均感染時間間隔及び 6 分未満の標準偏差であり，感染時間間隔は不規則に広がっていた．グループ B のパターンは，様々なボットネットによる攻撃が衝突したために，広く分散した可能性がある．

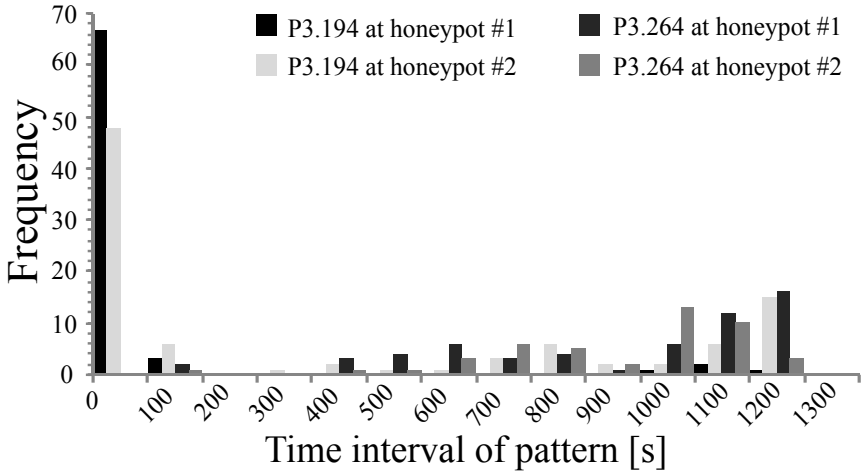


図 5.7: 系列感染パターン (3*length*) の感染時間間隔

5.4.5 PrefixSpan のパフォーマンス

CPU が *Intel*® *Core*™2 Duo T5750 2.00 GHz，OS が Ubuntu 10.10 with GCC version 4.4.5 のマシンを使用し，2*length* 及び 3*length* の系列感染パターンを抽出して，パフォーマンスを測る実験を行った．なお，PrefixSpan アルゴリズムは C++プログラミング言語で書かれている．

系列感染パターンを抽出時の PrefixSpan アルゴリズムの性能を図 5.8 に示す．実験のため，2 つのハニーポットのデータを前処理データとして，1 日，1 週間，1 ヶ月，3 ヶ月，6 ヶ月，1 年間のサイズで分割しており，それが X 軸である．Y 軸は一般的なパフォーマンステストとして，計算の経過時間 *s* を用いる．また，斜線は PrefixSpan のパフォーマンスを表す．結果は Honey003 と 004 でそれぞれ 47.058 bps，50.000 bps であった．

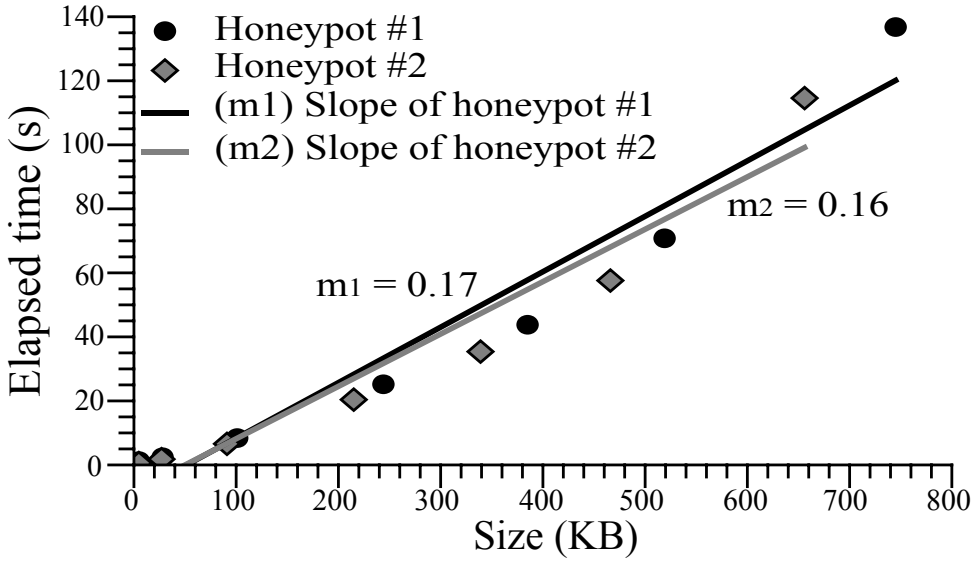


図 5.8: 系列感染パターン抽出時の PrefixSpan アルゴリズムのパフォーマンス

5.4.6 系列感染パターンのエントロピー解析

大規模データからデータマイニングを行うと，大量の価値あるパターンが抽出できる．例えば，本研究で用いた 94 台のハニーポットでは何千ものパターンが抽出されるが，ここで別の課題が発生する．すなわち,「本当に価値のあるパターンなのか？」である．

そこで，エントロピーの観点からパターンを分類することにより，どの攻撃パターンがコンピュータネットワークに対する攻撃として一般的なのかを判定し，連携感染パターンにおける重要な振る舞いを特定を試みる．エントロピー解析は，ネットワークセキュリティの分野で広く使用され，例えば，DDoS 攻撃の検出 [19] や異常にアクセスされた IP パケット [20]，パックされているマルウェアの実行ファイルの解析など様々な利用がされている [21].

エントロピーを求めるにあたって，各ハニーポットから抽出した数千の系列感染パターンを重複及び非重複に分類し，ダウンロード数の頻度が高い順番でソートした．

系列感染パターン S のエントロピーは次の式で定める．

$$H(S) = -\sum_{i=1}^{I} P(S_i) \log_2(P(S_i)), \tag{5.1}$$

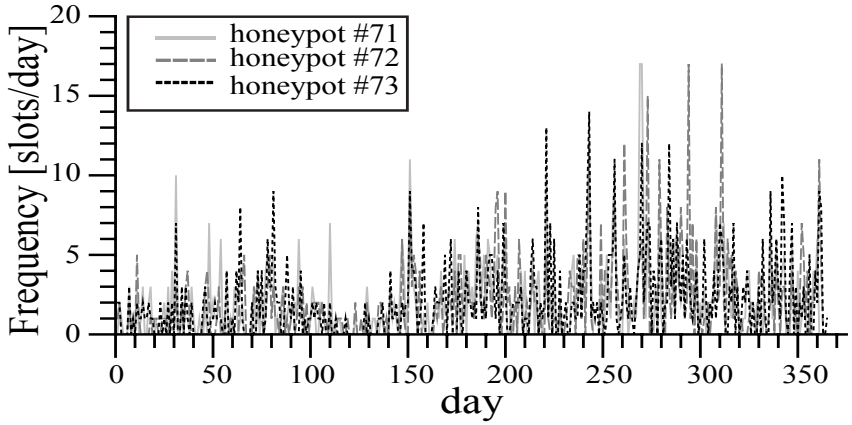
系列感染パターン S が i 番目のハニーポットを攻撃しようとする確率を $P(S_i)$，ハニーポット数を I と示す．このとき，各系列感染パターンがハニーポットを攻撃しようとする確率は同じである．例えば，系列感染パターン S に感染したハニーポット数が 10 であるとき，系列感染パターンの確率は $P(S_1) = P(S_2) = P(S_3) = ... = P(S_{10}) = 0.1$ である．また，最大のハニーポット数を $I = 94$ としたとき，エントロピーのスコアは $0 \leq H(S) \leq \log_2(94)$ の範囲となる．

エントロピーの計算結果を表 5.4 に示す．表の上部のグループは重複パターン，下部のグループは非重複パターンを表す．重複パターン $P_{3.1203}$ と非重複パターン $P_{3.194}$ は，それぞれ最大のエントロピーのスコアだった．これは，これらのパターンが最も多くのハニーポットに攻撃を試みたことを示している．

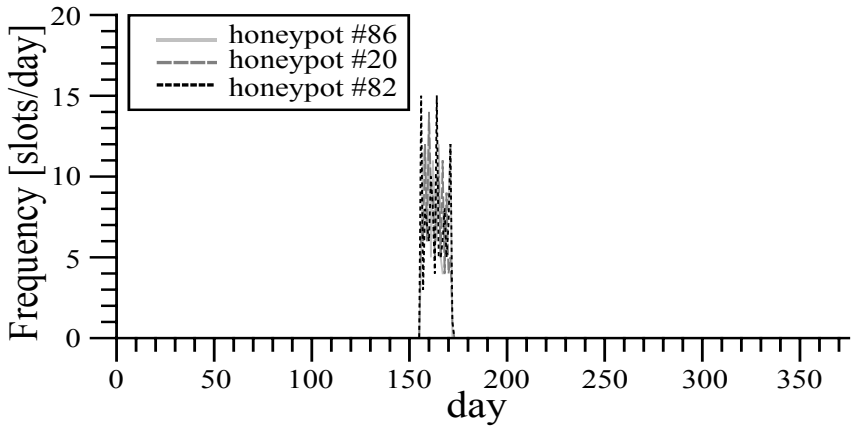
また，これら 2 つのパターンについて調査したところ，2 つのパターンで異なる特徴を持っていることが明らかになった．重複パターン $P_{3.1203}$ と非重複パターン $P_{3.194}$ の分布を図 5.9 に示す．図 5.9 (a) より，パターン $P_{3.1203}$ は 1 年間継続的に分布しているが，図 5.9 (b) のパターン $P_{3.194}$ は，短期間でかつ同じ日付に分布している．この結果から，パターン $P_{3.1203}$ の振る舞いは，このパターンに関わるマルウェアがいくつかのボットによって一般的に使用されていることを示唆している．逆に，パターン $P_{3.194}$ は多くのハニーポットで見られたが，図 5.9 (b) に示す通り特定の日付で，短い期間に感染する．すなわち，パターン $P_{3.194}$ は，特定の攻撃を目的として，特定のボットネットから送信されたパターンであると推定できる．

表 5.4: 全 94 台のハニーポットにおける系列感染パターンのエントロピー

ID	Pattern Name	Entropy
$P_{3.1203}$	PE_VIRUT.AV PE_VIRUT.AV PE_VIRUT.AV	6.0875
$P_{3.2425}$	TSPY_KOLABC.CH TSPY_KOLABC.CH TSPY_KOLABC.CH	5.9307
$P_{3.1590}$	PE_VIRUT.D-4 PE_VIRUT.D-4 PE_VIRUT.D-4	5.8826
$P_{3.857}$	PE_BOBAX.AK PE_BOBAX.AK PE_BOBAX.AK	5.8073
$P_{3.1463}$	PE_VIRUT.D-1 PE_VIRUT.D-1 PE_VIRUT.D-1	5.7814
⋮	⋮	⋮
$P_{3.2796}$	WORM_RBOT.GDJ WORM_RBOT.GDJ WORM_RBOT.GDJ	2.0
$P_{3.2528}$	TSPY_ONLINEG.TKJ TSPY_ONLINEG.TKJ TSPY_ONLINEG.TKJ	1.5850
$P_{3.2676}$	WORM_POEBOT.AKE TSPY_KOLABC.CH TSPY_KOLABC.CH	1.0
$P_{3.2611}$	WORM_KOLABC.BQ PE_VIRUT.YE WORM_KOLABC.BQ	0.0
$P_{3.1924}$	PE_VIRUT.YC PE_VIRUT.YC PE_VIRUT.YC	0.0
$P_{3.194}$	BKDR_RBOT.CZO WORM_HAMWEQ.AP TROJ_QHOST.WT	5.9307
$P_{3.242}$	BKDR_SCRYPT.ZHB BKDR_SDBOT.BU BKDR_VANBOT.HI	5.7279
$P_{3.2351}$	TROJ_QHOST.WT WORM_HAMWEQ.AP BKDR_POEBOT.AHP	5.6724
$P_{3.134}$	BKDR_POEBOT.GN TSPY_KOLABC.CH WORM_SWTYMLAI.CD	5.5849
$P_{3.1368}$	PE_VIRUT.AV TSPY_KOLABC.CH WORM_SWTYMLAI.CD	5.5546
⋮	⋮	⋮
$P_{3.635}$	BKDR_VANBOT.FM TROJ_PROXY.WE TROJ_PACK.DT	1
$P_{3.714}$	BKDR_VANBOT.LE TROJ_BUZUS.ADZ WORM_SPYBOT.ADS	1
$P_{3.2336}$	TROJ_QHOST.KY BKDR_RBOT.IA TROJ_VUNDO.MCS	0
$P_{3.2659}$	WORM_POEBOT.AKE BKDR_POEBOT.GN TSPY_KOLABC.CH	0
$P_{3.2641}$	WORM_PAKES.ABU PE_BOBAX.AK BKDR_VANBOT.LE	0



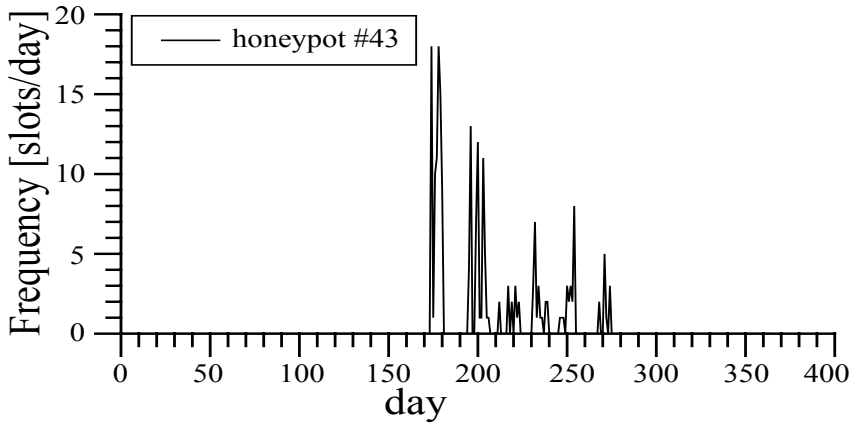
(a) 重複パターン $P_{3.1203}$



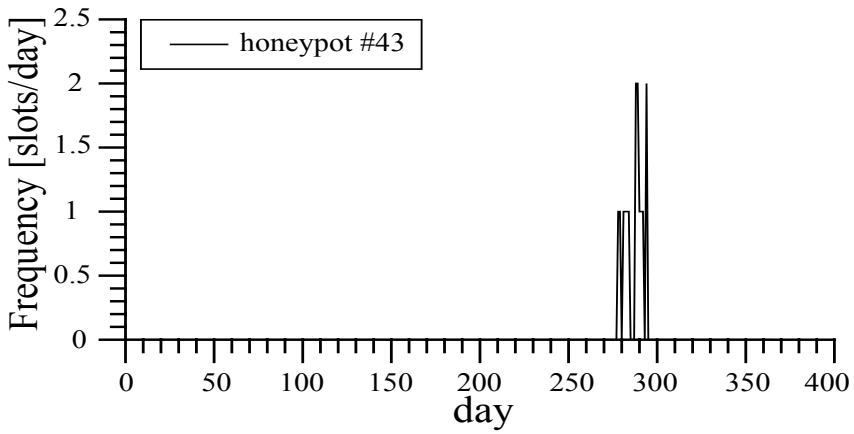
(b) 非重複パターン $P_{3.194}$

図 5.9: 複数のハニーポットで観測されたエントロピーのスコアが高い系列感染パターンの分布: (a) パターン $P_{3.1203}$，(b) パターン $P_{3.194}$

エントロピーのスコアが低いパターンとして，重複パターン $P_{3.1924}$ と非重複パターン $P_{3.2659}$ を図 5.10 に示す．いずれもエントロピーの値が低い．図 5.10 から，これらのパターンは単一のハニーポットでしか観測されていないことがわかる．そのため，これらのパターンはおそらくパターンの誤検出か，未経験のユーザによる偶発的な攻撃のいずれかであると考える．



(a) 重複パターン $P_{3.1924}$



(b) 非重複パターン $P_{3.2659}$

図 5.10: 単一のハニーポットで観測されたエントロピーのスコアが低い系列感染パターンの分布: (a) パターン $P_{3.1924}$，(b) パターン $P_{3.2659}$

5.5 まとめ

Apriori の欠点である順列を考慮しない点を PrefixSpan を用いることで解決できることを示した．連携感染を発見し，分析するために PrefixSpan は十分強力である．また，ボットネットによる攻撃の脅威をユーザーに警告するために役立ついくつかの振る舞いを明らかにした．連携感染は短期間内に複数の連続攻撃パターンによって実行されており，連携携感によって使用されるマルウェアはダウンロード時間か，配布サーバの送信元 IP アドレスに関する系列の特性を持っていた．エントロピー解析は，連携攻撃に関わる最も一般的な連続攻撃パターンを発見するのに有効である．

第6章
連携感染の変遷

6.1 概要

第4章にて，攻撃元データに対し，価値ある相関ルールを抽出するデータマイニング手法であるアソシエーション分析 (以下，Apriori) を適用し，関連性の強いマルウェアの組み合わせ，すなわち，連携感染を自動抽出する手法を提案した．また，第5章にて，系列パターンマイニングである PrefixSpan を用いることで，Apriori の欠点であった時系列を考慮したルールが抽出可能であることを示した．Apriori と PrefixSpan の違いを表 6.1 に示す.

表 6.1: データマイニング方法の比較

	Apriori	PrefixSpan
提案者	Agrawal，他 [11]	Pei，他 [12]
抽出対象	相関ルール (A，B → C)	シーケンスパターン (A，B，*，C)
精度	支持度，確信度	確信度
特徴	アイテムの集合 (順序なし)	シーケンス (順序あり)

しかし近年，新たに Gumblar をはじめとする Web 感染型マルウェアが台頭し，被害が増加している．逆に，攻撃元データに含まれるマルウェアの感染数は減少傾向にある．３年間のマルウェアのダウンロード数の推移を図 6.1 に示す.

これは攻撃の主流が Web 感染型マルウェアに移行して来ていることを示唆している．では，実際にマルウェアの連携感染は減少しているのだろうか.

そこで第6章では，研究用データセット CCC DATAset 2008 から 2010 の攻撃通信データ，攻撃元データを用いて，3年間に渡るマルウェアの振る舞いに着目し，データマイニング手法である Apriori と PrefixSpan を適用することで，連携感染の変遷を調査した．その結果明らかになった特徴を報告する.

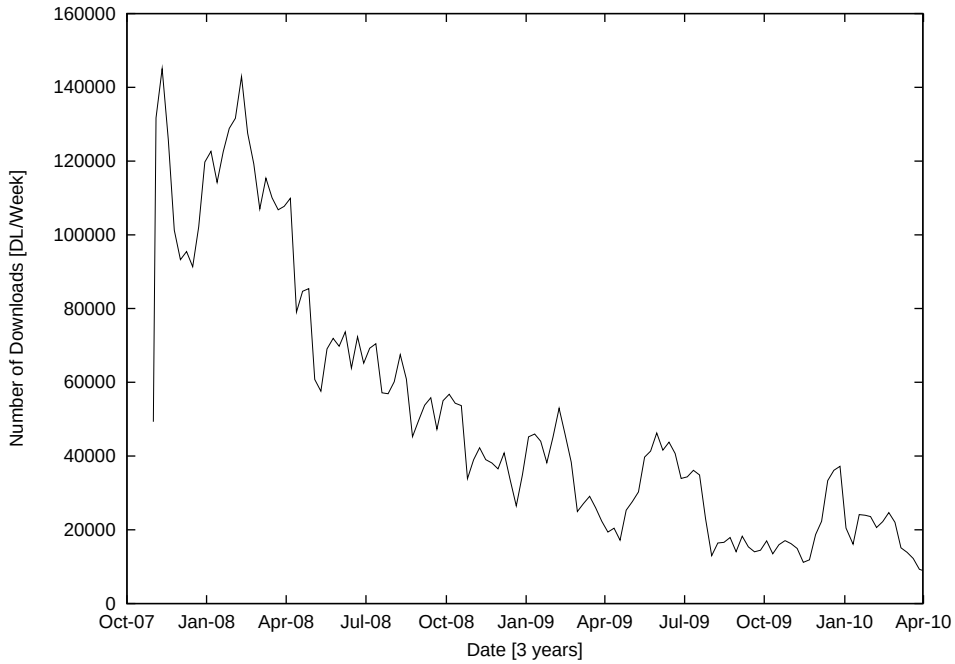


図 6.1: マルウェアのダウンロード数の推移

6.2 調査項目

第 6 章では，実験データとして，CCC DATAset 2008，2009，2010 の攻撃通信データと攻撃元データを使用する．

調査項目を以下に示す．

1. マルウェアの活動傾向
2. 連携感染の活動傾向
3. 連携感染が減少した理由

6.3 調査結果

6.3.1 マルウェアの活動傾向

攻撃元データから 3 年間観測されているマルウェアが存在するか調査した．結果を表 6.2 に示す．3 年間を通して上位のマルウェアは，PE_VIRUT.AV である．PE_VIRUT.AV は，3.1 節で述べた連携感染の起点となるマルウェアである．感染数は減少傾向だが，マルウェアファミリ名も PE が占めており，その勢力は衰えていない．

表 6.2: 3 年間共通して観測されたマルウェア

マルウェア名	2008 年		2009 年		2010 年	
	順位	Uniq.	順位	Uniq.	順位	Uniq.
PE_BOBAX.AK	8	47654	3	94324	32	8018
PE_VIRUT.AV	9	46741	2	222207	1	194557
WORM_ALLAPLE.IK	10	45033	12	30319	19	12564
PE_VIRUT.XV	20	26518	28	16625	31	8424
PE_VIRUT.XZ	46	14315	51	8885	33	7181
PE_VIRUT.PAU	63	10749	47	9347	21	11815
BKDR_VANBOT.HG	93	6050	43	11206	24	10404

次に，連携感染に用いられる IRC，DNS を攻撃通信データから抽出した．それぞれ表 6.3, 表 6.4 に示す．抽出には，PING の送信先を使い，数は各スロットのユニーク数としている．IRC では，3 年間共通して hub.*****.com が使用されていた．これは，いずれも PE を起点とした連携感染で用いられるドメインである．同様に，DNS でも一部ドメインが 3 年間使用されているのを確認できた (表中太字)．以上の結果から，連携感染はなくなっておらず，最低でも 1 つ以上のボットネットで使用されていると考えられる．

表 6.3: 3 年間共通して用いられた IRC サーバ

順位	2008 年		2009 年		2010 年	
	IRC ドメイン	数	IRC ドメイン	数	IRC ドメイン	数
1	hub.40***.com	81	hub.14***.com	35	pwned30.i***.net	31
2	i	38	-	-	pwned28.i***.net	30
3	hub.56***.com	36	-	-	hub.63***.com	23
4	hub.44***.com	31	-	-	hub.48***.com	20
5	aaa.59***.com	3	-	-	hub.27***.com	14
6	irc.foo***.com	2	-	-	no***.org	13
7	bl*.com	2	-	-	s*.com	8
8	FE7B03EC	1	-	-	ja**.org	5
9	F3B4433F	1	-	-	irc.fo***.fo	1

表 6.4: 2010 年の DNS ドメインとその比較

順位	DNS ドメイン	数	2008 年	2009 年
1	botz.noreta***.com	133		
2	proxim.ntkrn***.info	62		
3	checkip.dyn***.org	60		
4	www.whatism***.org	52		
5	tx.mostafaaljaaf***.net	35		
6	tx.nadersam***.org	32		
7	www.whatsmyipaddr***.com	31		
8	www.getm***.org	28		
9	**ss.ka***.com**	19	31 位	1 位
10	**ss.nadnad***.info**	16	81 位	5 位
11	**ss.MEMEH***.INFO**	15	90 位	
12	videogale***.com	12		
13	blah.swapixtr***.com	10		
26	xx.nadna***.info	2		

6.3.2 連携感染の活動傾向

本節では，攻撃元データから，Apriori，PrefixSpan を用いて連携感染を抽出し，連携感染の活動傾向を調査する．なお，2008 年の攻撃元データはハニーポットの記載がないため，使用していない．

まず，連携感染数の変化を図 6.2 に示す．全ハニーポット 730 日分のデータに対し，Apriori を用いて相関ルールの抽出を行い，月平均を求めた．Apriori を使用した理由は，PrefixSpan に比べ，抽出されるルール数が正確であったためである．系列の長さ 3 以上のパターンに絞っており，2 種類以下のマルウェア間のルール及び `UNKNOWN` を含むルールは除外している．図 6.2 から，マルウェアの減少に伴い，連携感染も減少傾向であることがわかる．

Honey001 ~ 010 をまとめたものを図 6.3 に示す．ハニーポットによって傾向の違いこそあるものの，全体の傾向としてはやはり減少傾向であった．攻撃通信データを解析した結果でも，2009 年は 3 種類の連携感染が確認できたことに対し，2010 年は 1 種類のみである．

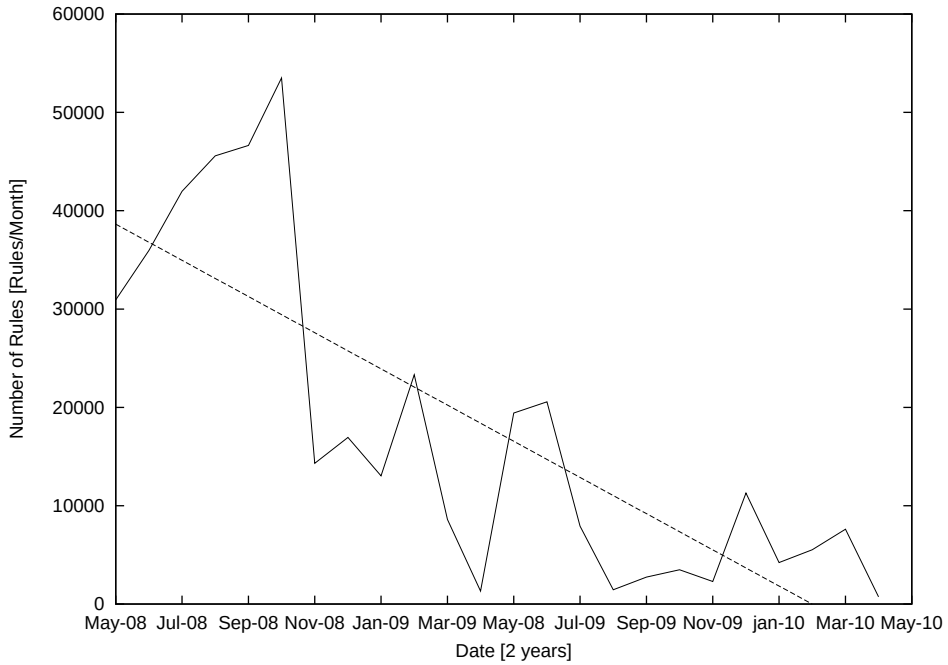


図 6.2: 2009，2010 年の連携感染数の推移 (全体)

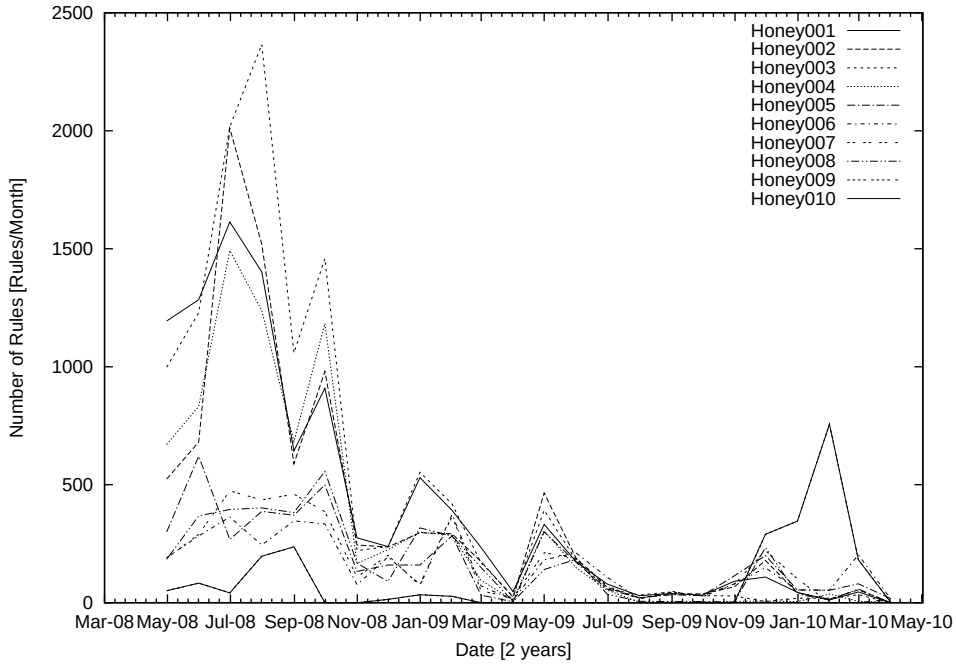


図 6.3: 2009，2010 年の連携感染数の推移 (Honey001 ~ 010)

次に，連携感染は何種類のマルウェアで構成された感染を行うのか，すなわち，平均連携マルウェア数を調査する．抽出には PrefixSpan を用いて連携マルウェア数は 3 以上とし，全ハニーポットの連携感染パターンを抽出して平均を求めた．PrefixSpan は時系列を考慮できるため，感染順を考慮して連携感染パターンを抽出することができるためである．こうして求められた平均連携マルウェア数の変化を図 6.4 に示す．連携感染総数の減少とは逆に，構成マルウェア数は増加している．これは連携感染はより複雑化し，巧妙化してきていることを示している．

また，攻撃通信データを解析したところ，HTTP GET でダウンロードされるマルウェア数は，2008，2009 年で 2 種類であったのに対し，2010 年では 5 種類となっており，ここからも連携パターンの複雑化が裏づけられる．

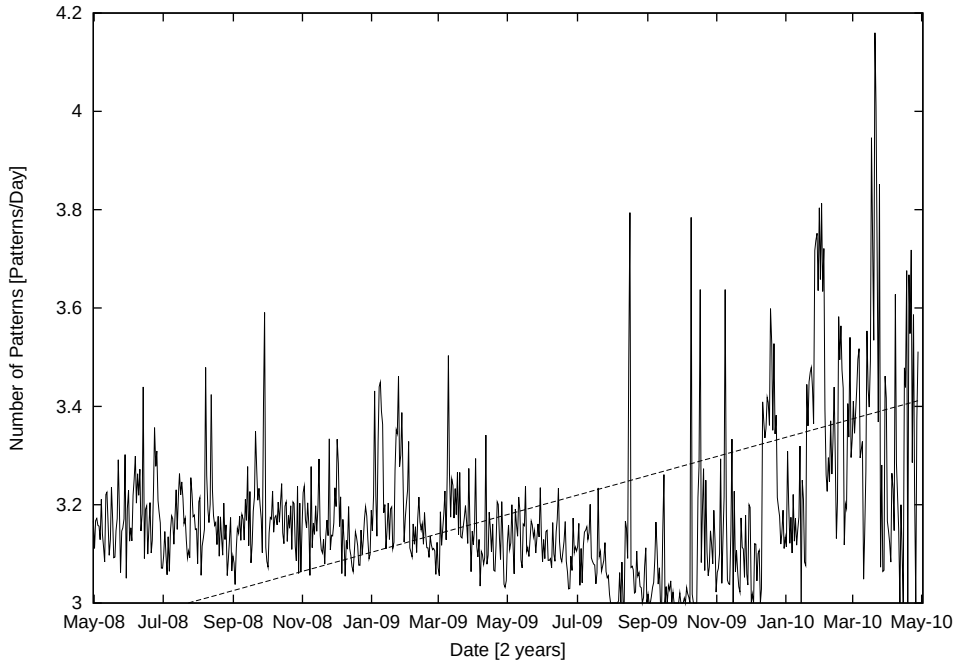


図 6.4: 連携感染の平均連携マルウェア数

6.3.3 連携感染が減少した理由

本節では，なぜダウンロード数，連携感染数が減少したのかを考察する．

CCC DATAset 2010 の攻撃元データ数が不自然に少なく，検出誤りが生じていると考えた．そこで，攻撃通信データを解析し，各スロットを TCP と UDP の通信に分割し，それぞれ tcpflow，TFTPgrab を用いて分析することでマルウェア検体を得て，攻撃元データとの比較を行った．表 6.5 に比較結果を示す[1)]．

表 6.5: 2010 年の攻撃通信データと攻撃元データの比較 (一部)

No.	マルウェア名	Prot.	攻撃通信	攻撃元	誤差
1	WORM_DOWNAD.AD	TCP	118	79	−39
2	WORM_PALEVO.SMD	TCP	106	36	−70
3	WORM_PALEVO.BL	TCP	49	12	−37
4	PE_VIRUT.AV	TCP	42	26	−16
5	TROJ_BUZUS.MC	TCP	25	11	−14
6	BKDR_RBOT.SMA	UDP	43	13	−30
7	BKDR_MYBOT.AH	UDP	13	4	−9
8	WORM_SDBOT.CEM	UDP	6	5	−1
9	WORM_MYTOB.IR	UDP	1	0	−1
合計	-	-	512	261	−251

表 6.5 より，攻撃通信データに含まれるマルウェアが攻撃元データに含まれていない．特に TCP 通信では WORM_PALEVO.SMD，UDP 通信では BKDR_RBOT.SMA が数多く得られたが，攻撃元データとの差はそれぞれ 70 個と 30 個と大きい．

以上より，2010 年の攻撃元データの 261 件には，攻撃通信データの 512 件に対して，1.961 倍の未検出があったことがわかる．2009 年では，攻撃元データが 200，攻撃通信データが 221 で 1.105 倍の未検出に留まっていた．この結果を考慮して，図 6.1 を補正したダウンロード数の推移を図 6.5 に示す．

[1)] 3 年間の攻撃元データ及び Virus Total に該当するハッシュ値がないマルウェア検体は調査対象から外した．なお，2008 年の攻撃通信データから分析して得られたマルウェア数は 673 個で圧倒的に多いが，攻撃元データと比較できないため省略する．

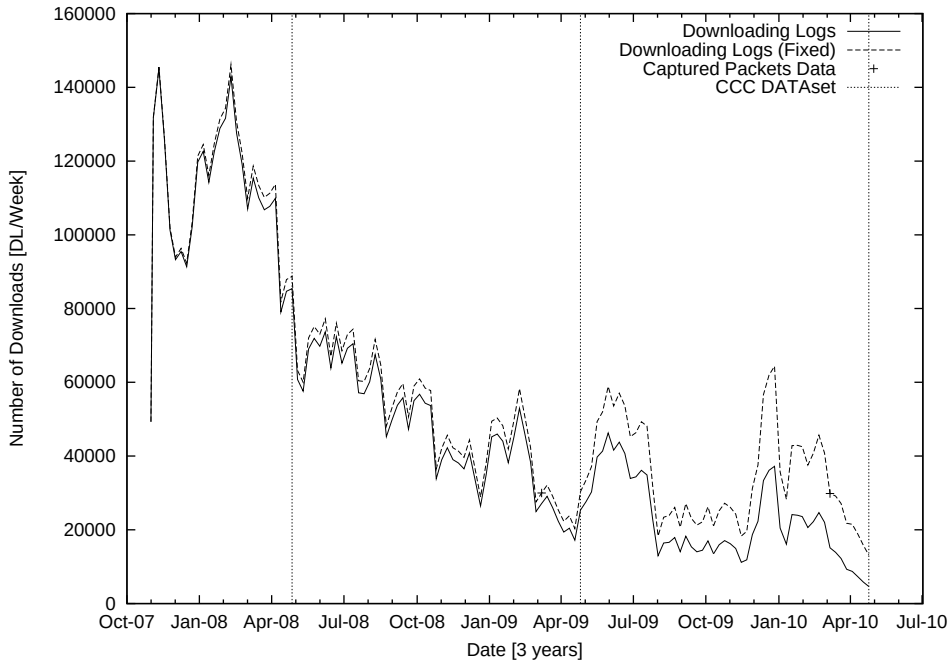


図 6.5: 補正したマルウェアのダウンロード数の推移

実際の未検出数がわかる 2009 年，2010 年の攻撃通信データを使用して，その値を元に補正を行っている．ダウンロード数は全ハニーポットにおける総数である．なぜならば，2008 年は，攻撃元データにハニーポット ID が含まれず，区別できないためである．なお，2009 年 5 月末で 2 台のハニーポットが停止している．

全体の傾向として，HTTP GET によってダウンロードされたマルウェア，あるいは，UDP 通信によるマルウェアが未検出であることが多かった．同一のマルウェアを検出できていることもあったため，マルウェア検体自体が原因，もしくは，収集しているハニーポット固有の問題と考えられる．

6.4 まとめ

過去 3 年間の攻撃通信データ，攻撃元データを用いて，連携感染の変遷及び特徴を報告した．連携感染数が減少する一方で，感染するマルウェアの連携パターン数は増加していた．この増加は，2010 年の攻撃通信データ，攻撃元データの解析結果から裏づけられた．

第7章

AprioriとPrefixSpanによる連携感染のハイブリッド検出手法

7.1 AprioriとPrefixSpanの比較

実際に観測されたデータセットを用いて，マルウェアの連携感染を検出するために，2つの自動化アルゴリズムであるAprioriとPrefixSpanを評価する．2009年2月の実験結果の一部を表7.1に示す．この結果は，実際にトレンドマイクロ社から報告されているTSPY_KOLABC.CH, WORM_SWTYMLAI.CD，BKDR_POEBOT.Gの連携感染を抜粋した結果である[22].

ここで，Apriori のSlotsは，発見したタイムスロット数の頻度を表し，手動調査によって得られた真のタイムスロット数をTrue [Slots]で示す．同様に，PrefixSpanのPtnsは検出した系列感染パターン数を表し，True [Ptns]は真のパターン数を示している．すなわち，Trueは評価するにあたり，実験データから得た解答である．

また，原則として，Apriori の頻度はPrefixSpanのものより大きい．一方，PrefixSpanによって検出されたいくつかの系列パターンはAprioriのカラムには表記されていない．これは，PrefixSpanでは別々に抽出されたパターンだが，Apriori では感染順序を考慮せず，同じパターンとして集計しているためである．

表7.1: AprioriとPrefixSpanの比較

日付	Apriori			PrefixSpan		
	相関ルール (集合)	Slots	True [Slots]	系列パターン	Ptns	True [Ptns]
2009/02/03	WORM, BKDR ⇒ TSPY	4	4	TSPY ⇒ WORM ⇒ TKDR	3	9
2009/02/04	BKDR, TSPY ⇒ WORM	14	14	TSPY ⇒ BKDR ⇒ WORM	3	29
				TSPY ⇒ WORM ⇒ BKDR	7	
				WORM ⇒ BKDR ⇒ TSPY	4	
				WORM ⇒ TSPY ⇒ BKDR	12	
			⋮			
2009/02/28	BKDR, TSPY ⇒ WORM	7	7	TSPY ⇒ WORM ⇒ BKDR	5	14
	BKDR, WORM ⇒ TSPY	7		WORM ⇒ TSPY ⇒ BKDR	3	
合計		464	315		482	575

表 7.1 の 2 月 4 日を例として説明する．Apriori は，スロット単位で確実に TSPY_KOLABC.CH, WORM_SWTYMLAI.CD，BKDR_POEBOT.G の連携感染を抽出している．これは 2 月 4 日の 72 スロットのうち，この連携感染を含むスロットが 14 スロットあることを示す．しかし，Apriori は 2 月 28 日では 7 スロットの誤検出が確認できる．これは X と Y の違いから，実際には同じ集合であるパターンを別のパターンと認識して抽出してしまうためである．Apriori ではこのような誤検出が発生することがあるので注意が必要だが，同じパターンとして無視すれば問題ない．

一方，PrefixSpan は Apriori よりも False Positive は低いが，False Negative は高い．すなわち，未検出が発生する．例えば，2 月 4 日より，PrefixSpan では 4 種類のパターンをパターン単位で検出に成功している．しかし，3 種類のマルウェアによる考えられる組み合わせ全 6 パターンのうち，BKDR を頭とするパターンは頻度が低いために枝狩りされ，検出されていない．

以上より，Apriori は連携感染を含むタイムスロットの抽出，PrefixSpan は連携感染パターンの抽出に適している．連携感染がどのマルウェアで構成されるかなど，感染順序を考慮せず，単に連携感染と思われる組み合わせを発見したい場合は，Apriori が有効であり，順序を考慮した正確なパターンを求めたいときは，PrefixSpan を使用すると良い．

7.2 検出精度

表 7.1 を元に Apriori と PrefixSpan の検出精度を纏めたものを表 7.2，表 7.3 に示す．Apriori はタイムスロット，PrefixSpan はパターンに対してそれぞれ精度を求めた．支持度は Apriori が 3 スロット，PrefixSpan がパターンの頻度を 3 回 以上と設定している．Apriori は 464 スロットのうち，149 の 誤検出をしているが，未検出はない．PrefixSpan は誤検出はないが，575 パターンのうち，93 パターンの検出に失敗している．

また，表 7.2，表 7.3 から再現率，適合率を求めた結果を表 7.4 に示す．この結果より，再現率が高く，適合率が低い Apriori はスロット単位で連携感染を抽出できるが，関係のないスロットも抽出される．逆に再現率が低く，適合率が高い PrefixSpan はパターンを正確に抽出できるが，頻度の低いパターンを取りこぼす．つまり，最小支持度の設定を上げすぎるとパターンが抽出できなくなるため，適切な支持度を設定する必要がある．

表 7.2: Accuracy in Apriori

	連携	非連携	合計
検出	315	149	464
未検出	0	N/A	N/A
合計	315	149	464

表 7.3: Accuracy in PrefixSpan

	連携	非連携	合計
検出	482	0	482
未検出	93	N/A	93
合計	575	N/A	575

表 7.4: Recall and Precision

	Apriori	PrefixSpan	提案方式
再現率	315/315 = 1	482/575 = 0.838	545/575 = 0.947
適合率	315/464 = 0.678	482/482 = 1	482/482 = 1

7.3 Apriori と PrefixSpan のハイブリッド検出方式

調査結果より，2 つの自動化アルゴリズムを組み合わせることにより，それぞれの欠点を補い，さらなる精度の向上ができると考えた．そこで，Apriori と PrefixSpan のハイブリッド検出方式を提案する．

7.1 節と同様に，表 7.1 の 2 月 4 日のデータを例に説明する．Apriori では 1 種類に纏められている相関ルールが PrefixSpan では 4 種類の系列パターンとして検出されている．しかし，実際には上記の 3 種類の組み合わせだけでなく，その他の連携感染も検出される．なぜならば，表 7.1 ではトレンドマイクロ社の報告により，TSPY_KOLABC.CH，WORM_SWTYMLAI.CD，BKDR_POEBOT.GN の連携感染を抜粋できたが，実際には最初から連携感染を特定することはできないためである．例えば，Apriori では 4 種類のルール，PrefixSpan では 32 種類のパターンが検出され，一見して連携感染を判断するのは難しい．ハイブリッド検出方式の具体的な手順を以下に示す．

1. Apriori アルゴリズムの適用

Apriori 2 月 4 日の実験データに対して，Apriori を適用し，連携感染と考えられる相関ルールを抽出する．支持度は 5 スロット以上，確信度は 80 %以上である．Apriori の出力結果を以下に示す．結果は “相関ルール名 (スロット数，確信度)” を表している．

1. BKDR_POEBOT.GN ⇒ TSPY_KOLABC.CH (16，87.5)
2. BKDR_POEBOT.GN ⇒ WORM_SWTYMLAI.CD (16，100.0)
3. WORM_SWTYMLAI.CD ⇒ BKDR_POEBOT.GN (19，84.2)
4. TSPY_KOLABC.CH ⇒ WORM_SWTYMLAI.CD (19，89.5)
5. WORM_SWTYMLAI.CD ⇒ TSPY_KOLABC.CH (19，89.5)
6. PE_VIRUT.AV，BKDR_POEBOT.GN ⇒ TSPY_KOLABC.CH (8，87.5)
7. PE_VIRUT.AV，BKDR_POEBOT.GN ⇒ WORM_SWTYMLAI.CD (8，100.0)
8. PE_VIRUT.AV，WORM_SWTYMLAI.CD ⇒ BKDR_POEBOT.GN (10，80.0)
9. PE_VIRUT.AV，TSPY_KOLABC.CH ⇒ WORM_SWTYMLAI.CD (9，100.0)
10. PE_VIRUT.AV，WORM_SWTYMLAI.CD ⇒ TSPY_KOLABC.CH (10，90.0)
11. BKDR_POEBOT.GN，TSPY_KOLABC.CH ⇒ WORM_SWTYMLAI.CD (14，100.0)

12. BKDR_POEBOT.GN, WORM_SWTYMLAI.CD ⇒ TSPY_KOLABC.CH (16, 87.5)

13. TSPY_KOLABC.CH, WORM_SWTYMLAI.CD ⇒ BKDR_POEBOT.GN (17, 82.4)

14. PE_VIRUT.AV, BKDR_POEBOT.GN, TSPY_KOLABC.CH ⇒ WORM_SWTYMLAI.CD (7, 100.0)

15. PE_VIRUT.AV, BKDR_POEBOT.GN, WORM_SWTYMLAI.CD ⇒ TSPY_KOLABC.CH (8, 87.5)

出力結果より，3 種類のマルウェアで構成される相関ルールを抜粋する．ルールはアイテムの集合なのでユニークとすると，以下の 4 種類のルールが発見できる．

6. PE_VIRUT.AV, BKDR_POEBOT.GN ⇒ TSPY_KOLABC.CH (8, 87.5)
   ⇒ A. PE_VIRUT.AV, BKDR_POEBOT.GN, TSPY_KOLABC.CH

7. PE_VIRUT.AV, BKDR_POEBOT.GN ⇒ WORM_SWTYMLAI.CD (8, 100.0)

8. PE_VIRUT.AV, WORM_SWTYMLAI.CD ⇒ BKDR_POEBOT.GN (10, 80.0)
   ⇒ B. PE_VIRUT.AV, BKDR_POEBOT.GN, WORM_SWTYMLAI.CD

9. PE_VIRUT.AV, TSPY_KOLABC.CH ⇒ WORM_SWTYMLAI.CD (9, 100.0)

10. PE_VIRUT.AV, WORM_SWTYMLAI.CD ⇒ TSPY_KOLABC.CH (10, 90.0)
    ⇒ C. PE_VIRUT.AV, TSPY_KOLABC.CH, WORM_SWTYMLAI.CD

11. BKDR_POEBOT.GN, TSPY_KOLABC.CH ⇒ WORM_SWTYMLAI.CD (14, 100.0)

12. BKDR_POEBOT.GN, WORM_SWTYMLAI.CD ⇒ TSPY_KOLABC.CH (16, 87.5)

13. TSPY_KOLABC.CH, WORM_SWTYMLAI.CD ⇒ BKDR_POEBOT.GN (17, 82.4)
    ⇒ D. BKDR_POEBOT.GN, TSPY_KOLABC.CH, WORM_SWTYMLAI.CD

2. PrefixSpan アルゴリズムの適用

同様に，実験データに対し，PrefixSpan を適用する．全てのパターンを網羅するため，支持度は低く設定する必要がある．今回は支持度を 3 とする．PrefixSpan の出力結果を以下に示す．回数は系列パターンの頻度数を表す．また，パターンは 32 種類出力された．

1. BKDR_POEBOT.GN TSPY_KOLABC.CH BKDR_POEBOT.GN 3 回
2. BKDR_POEBOT.GN WORM_SWTYMLAI.CD BKDR_POEBOT.GN 3 回
3. BKDR_SCRYPT.ZHB BKDR_SCRYPT.ZHB BKDR_POEBOT.GN 3 回
4. BKDR_SCRYPT.ZHB BKDR_SCRYPT.ZHB TSPY_KOLABC.CH 3 回
5. BKDR_SCRYPT.ZHB BKDR_SCRYPT.ZHB WORM_SWTYMLAI.CD 3 回
6. BKDR_SCRYPT.ZHB TSPY_KOLABC.CH BKDR_POEBOT.GN 3 回
7. BKDR_SCRYPT.ZHB WORM_SWTYMLAI.CD BKDR_POEBOT.GN 3 回
8. BKDR_SCRYPT.ZHB WORM_SWTYMLAI.CD TSPY_KOLABC.CH 3 回
9. PE_VIRUT.AV PE_VIRUT.AV PE_VIRUT.AV 3 回
10. PE_VIRUT.AV PE_VIRUT.AV TSPY_KOLABC.CH 3 回
11. PE_VIRUT.AV PE_VIRUT.AV WORM_SWTYMLAI.CD 3 回
12. PE_VIRUT.AV TSPY_KOLABC.CH PE_VIRUT.AV 3 回
13. PE_VIRUT.AV WORM_SWTYMLAI.CD PE_VIRUT.AV 3 回
14. TSPY_KOLABC.CH BKDR_POEBOT.GN BKDR_POEBOT.GN 3 回
15. TSPY_KOLABC.CH BKDR_POEBOT.GN WORM_SWTYMLAI.CD 3 回
16. TSPY_KOLABC.CH TSPY_KOLABC.CH BKDR_POEBOT.GN 3 回
17. TSPY_KOLABC.CH WORM_SWTYMLAI.CD TSPY_KOLABC.CH 3 回
18. WORM_SWTYMLAI.CD BKDR_POEBOT.GN BKDR_POEBOT.GN 3 回
19. WORM_SWTYMLAI.CD BKDR_POEBOT.GN WORM_SWTYMLAI.CD 3 回
20. WORM_SWTYMLAI.CD TSPY_KOLABC.CH TSPY_KOLABC.CH 3 回
21. WORM_SWTYMLAI.CD WORM_SWTYMLAI.CD TSPY_KOLABC.CH 3 回

22. PE_VIRUT.AV TSPY_KOLABC.CH WORM_SWTYMLAI.CD 4 回

23. TSPY_KOLABC.CH BKDR_POEBOT.GN TSPY_KOLABC.CH 4 回

24. WORM_SWTYMLAI.CD BKDR_POEBOT.GN TSPY_KOLABC.CH 4 回

25. WORM_SWTYMLAI.CD TSPY_KOLABC.CH PE_VIRUT.AV 4 回

26. WORM_SWTYMLAI.CD TSPY_KOLABC.CH WORM_SWTYMLAI.CD 4 回

27. WORM_SWTYMLAI.CD WORM_SWTYMLAI.CD BKDR_POEBOT.GN 5 回

28. PE_VIRUT.AV WORM_SWTYMLAI.CD TSPY_KOLABC.CH 6 回

29. PE_VIRUT.AV TSPY_KOLABC.CH BKDR_POEBOT.GN 7 回

30. PE_VIRUT.AV WORM_SWTYMLAI.CD BKDR_POEBOT.GN 7 回

31. TSPY_KOLABC.CH WORM_SWTYMLAI.CD BKDR_POEBOT.GN 7 回

32. WORM_SWTYMLAI.CD TSPY_KOLABC.CH BKDR_POEBOT.GN 12 回

相関ルール D. BKDR_POEBOT.GN，TSPY_KOLABC.CH，WORM_SWTYMLAI.CD より，PrefixSpan の結果を抽出する．結果として，32 種類から以下の 4 種類の系列パターンまで削減できる．

15. TSPY_KOLABC.CH BKDR_POEBOT.GN WORM_SWTYMLAI.CD 3 回

24. WORM_SWTYMLAI.CD BKDR_POEBOT.GN TSPY_KOLABC.CH 4 回

31. TSPY_KOLABC.CH WORM_SWTYMLAI.CD BKDR_POEBOT.GN 7 回

32. WORM_SWTYMLAI.CD TSPY_KOLABC.CH BKDR_POEBOT.GN 12 回

以上より，BKDR_POEBOT.GN，TSPY_KOLABC.CH，WORM_SWTYMLAI.CD の連携感染は，以下の頻度が高かったいずれかの順列で感染している可能性が高い．

31. TSPY_KOLABC.CH WORM_SWTYMLAI.CD BKDR_POEBOT.GN 7 回

32. WORM_SWTYMLAI.CD TSPY_KOLABC.CH BKDR_POEBOT.GN 12 回

この結果は，31 番目と 32 番目の系列パターンがボットネットに使用されている可能性が高いことを示唆する．また，このように連携感染を特定するためには，観測された多くの無関係のマルウェアに対処しなければならない．

7.4 まとめ

最後に，Apriori と PrefixSpan を組み合わせることにより得られる利点を以下に示す.

(1) 正確な順列及び頻度の連携感染を検出可能

特定した系列パターンの頻度を合計すると 3 + 7 + 4 + 12 = 26 回となる．それぞれのパターンの頻度としては確かに正しいが，このパターンが 1 つの攻撃方法である仮定したとき，単純に合計して 26 回とするのは間違いである．理由は，PrefixSpan では同一スロット内で 2 つ以上のパターンが考えられる場合，それぞれ別のパターンと認識し，重複してカウントを行うためである．しかし，Apriori を使用すれば正しい結果に修正できる．表 7.1 に示す通り，これらのパターンが現れた 14 スロットが正しい回数となる．これより，正確な順列及び頻度の連携感染を特定できる.

(2) 連携感染の未検出を防止可能

Apriori の出力結果から，4 種類の相関ルールが 8 スロット以上で観測されている．そのため，PrefixSpan でも同様の支持度，すなわち，8 を設定すれば，これらのルールを含む系列パターンを抽出できるように思えるが，残念ながら，PrefixSpan では相関ルール A，B，C で構成されるパターンが全て未検出となる．これは，PrefixSpan によって抽出される各パターンが 7 以下の頻度しかなかったためである．これが 7.2 節にて，PrefixSpan では高い支持度を設定できないとした理由であり，Apriori でまず，連携感染の組み合わせを特定する理由でもある．また，Apriori で先に連携感染を検出することにより，PrefixSpan では枝刈りを考慮する必要がなくなるため，支持度を一般的な値より低くすることが可能である．ハイブリッド検出方式の精度を表 7.4 に示す．Apriori を適用後に PrefixSpan を適用することを前提とし，PrefixSpan の支持度を 3 から 2 に下げて精度を求めている．結果として，未検出がなくなり，再現率が向上した.

Apriori と PrefixSpan を用いたハイブリット検出方式を提案した．2 つのアルゴリズムを組み合わせることにより，それぞれの欠点を補完することができ，精度が向上した．提案手法は，連携感染を効率良くかつ正確に特定できる.

第8章
関連研究と応用の可能性

8.1 関連研究

ボットネットの振る舞いに関して様々な観点からボットネットの特徴が報告がされている．本章では，ハニーポットによって収集したデータを使用した研究に焦点をあて，(1) 発見的手法，(2) N-gram，(3) クラスタリング，(4) 主成分分析 (PCA) の 4 つのカテゴリに分類し，関連研究を述べる．

8.1.1 発見的手法

ボットネットの連携感染に関するマルウェアを検出する発見的手法が提案されている [9]．この研究は，ボットネットの攻撃間の特徴を決定する有用な情報を提供する．ただし，発見的なアプローチでは，新しい攻撃に対して適用できない欠点がある．

8.1.2 *N-gram* アルゴリズム

Lu Wei らによって，ボットネットのコミュニティを自動で調査する研究が行われている [23]．この研究は，大規模な Wi-Fi ネットワークにおける正常なネットワークトラフィックを調査するために，cross-association クラスタリングアルゴリズムと *N-gram* アルゴリズムを使用する．彼らは，正常なネットワークアプリケーションのグループを分類する自動アプリケーションと人間とボットネットによる悪意ある活動を区別するための一般的な手法を実現できるとしている．

目標を達成するためには以下の 3 つの手順が必要である．まず，未知のフローを得るために，ペイロード署名に基づく入力ネットワークのフローを分類する．次に，クロスアソシエーションクラスタリングに基づいて，未知のフローを調査し，アプリケーションのグループを分類する．最後に，ネットワークのフローが人間やボットによるものかどうかを定義するために，*N-gram* アルゴリズムを使用する．

8.1.3 トラフィックフローのクラスタリング

Thonnard と Dacier はグラフベースのクラスタリングを用いて，時系列の類似性という観点から，ハニーポットで収集したインターネットのネットワークトラフィックに関するデータセットを調査した [18]．時系列分析による適切な類似性測度によって，いくつかのワームやボットネットの活動を識別できることを発見している．

他のアプローチとしては，Gu らの BotMiner がある [24]．この提案手法は活動と通信という 2 つの観点からクラスタリングを行い，その結果を元に相関関係を求めることで，ボットネットの一部である感染コンピュータのグループを検出する．まず，ホストの悪意ある活動の種類や特徴に応じた分類と無関係な通信データをフィルタリングした後に，類似した通信のフローを共有しているホストを分類を行い，ネットワークトラフィックをクラスタリングする．そして，ホストが両方のクラスタに属していた場合，ボットネットの一部であると識別する．ただし，この手法はボットへの感染防止は考慮しておらず，ボットネットによる攻撃の早期検出のためには設計されていない．

8.1.4 Principal Component Analysis (PCA)

Husna らはスパムメールにおける類似性，特に時間的な特徴に着目し，スパマーの行動パターンを調査している [25]．主成分分析は，活動時間や Content-Length，メールの送信頻度などの特徴から最も情報量の多い特徴，すなわち，どの特徴が重要かを特定するために適用する．また，振る舞いの類似性に基づいてクラスタリングを適用し，ボットネットのグループにおけるスパム送信者を分類する．彼らはこの手法により，スパムドメインの類似する振る舞いから，正確にボットネットを識別できるとしている．

しかし，ボットネットは，攻撃者がマルウェアを送信し，感染した多くのインターネットに接続しているコンピュータを収集することで構築される．すなわち，この手法はボットネットによって生成されたスパムに対してのみ有効であり，スパムの振る舞いからボットネットの正確な特定は難しいと考える．これはボットネットの構築を予測するには，ネットワーク自身があまり効果的ではないことを意味している．

一方，本論文で示した攻撃者によって送信された時間に基づく連携感染の分類は，ボットネットの構築の初日を特定することが可能である．すなわち，初期の段階からボットネットの脅威を予測でき，ボットネットから派生する新たな脅威を防ぐことができる．

8.1.5 考察

これらの関連研究が連携感染を特定できない理由として，様々な属性に関する特定の連続パターンが挙げられる．例えば，マルウェア名 (ハッシュ値)，送信元 IP アドレス，マルウェアが感染する時間間隔などがある．実際に，これまで示した通り，ボットネットの攻撃は連続的に確立されている．重要な点は，ハニーポットによって連続的にかつリアルタイムでダウンロードされる**マルウェアの連続したパターン**を考慮することにある．

上記のアプローチ [23, 18, 24, 25] では，主にボットネットによる攻撃を詳細に調査することを目的として，ネットワークのフローを様々な評価基準，例えば，アプリケーションのグループ，送信元や送信先 IP アドレス，活動時間，Content-length，類似した悪意ある活動などの基準を元に分類し，クラスタリングを適用していた．

しかし，クラスタリングは特定の評価基準の類似性に焦点をあてて行われ，悪意ある活動の順序は十分に考慮していない．以上の理由から，クラスタリングでは新しい悪意ある活動やボットネットの攻撃を検出するのは困難である．

8.2 応用の可能性

近年のネットワークセキュリティ分野では，インターネット上における脅威を解決する様々な手法が提案されている．それらの手法に対し，本論文の提案手法を応用できる可能性がある手法を以下に示す.

(1) 侵入検知防止システム (IDPS)

提案手法の結果から，新しいネットワークベースの侵入検知システムを考える．ボットネットは多くの種類の攻撃を発生させるため，特定の IDPS 技術では全ての攻撃に対応するのは難しい．そこで，ハニーポットによってダウンロードされたマルウェアの連携感染パターンとモニタリングを行うことによって，不振な振る舞いを特定し，広範囲をカバーする IDPS を提供する．これにより，ボットネットの予測が可能になれば，ボットネットによる多くの種類の攻撃を排除できるようになる．

(2) ボットネットファイアウォール

図 5.6 で示した分類と振る舞いは，攻撃のアラートなど価値ある情報の提供を可能にする．攻撃者が感染したコンピュータを使用し，ボットネットとして確立するには，約 20〜25 日間を必要とする．すなわち，攻撃者による感染が行われた初日を特定できれば，進行中の脅威を防ぐための行動を取ることが可能である．そこで，定期的にデータマイニングを行い，第 5 章 5.4.4 節の図 5.6 と同じように，リアルタイムの統計情報を得て，ダウンロード数の頻度を監視する．これにより，最初の感染日を特定することができ，ボットネットの起点と考えられる発信源を遮断できる可能性がある．

(3) ボットネットトラッキング

ボットネットは連携した攻撃方法を用いるため，ハニーポットによってダウンロードされたマルウェアのシーケンスは，第 5 章 sec-3-2-3 節の図 5.4 に示したように，特定の連携パターンの形式を持っている．これはボットネットを追跡する上で，特定の攻撃を明らかにする貴重な情報である．また，マルウェアの連携感染パターンを特定することで，どの手順で被害者のコンピュータを危険に晒すのか，すなわち，ボットネットの戦略を明らかにできる．特に連携感染パターンに該当する送信元 IP アドレスは，ネットワーク上で行われた不正行為を立証する手助けとなる．

第9章　結論と今後の課題

9.1　結論

本論文では，2 つの自動検出アルゴリズム Apriori と PrefixSpan を用いた連携感染を機械的に抽出する方式を提案し，通信データを詳細に分析して得られる結果とほぼ同じ結果が抽出できることを実証した．また，Apriori と PrefixSpan の利点と欠点を調査し，それぞれを組み合わせることにより，2 つのアルゴリズムの欠点を補完して効率良くかつ正確に連携感染を特定できるハイブリッド検出方式を提案した．また，これらの手法を使用し，連携感染の解析を行い，以下のことがわかった．

Apriori アルゴリズム

TROJ と WORM の関連性が強い相関ルールが最も頻出していた．広範囲で観測されたルールは，連携感染を行っている可能性が高い事を示し，長期間で観測した結果では，連携感染期間は短い事が分かった．

PrefixSpan アルゴリズム

ボットネットによる攻撃の脅威をユーザーに警告するために役立ついくつかの振る舞いを明らかにした．連携感染は短期間内に複数の連続攻撃パターンによって実行されており，連携携感によって使用されるマルウェアはダウンロード時間か，配布サーバの送信元 IP アドレスに関する系列の特性を持っていた．エントロピー解析は，連携攻撃に関わる最も一般的な連続攻撃パターンを発見するのに有効である．

連携感染の変遷

過去 3 年間の攻撃通信データ，攻撃元データを用いて，連携感染の変遷及び特徴を報告した．連携感染数が減少する一方で，感染するマルウェアの連携パターン数は増加していた．

9.2 今後の課題

今後の課題として，提案手法を用いて連携感染の動向について今後も調査することや第8章で示した新しい侵入検知防止システム，ボットネットの送信元を遮断する新しいボットネットファイアウォール，そして，送信されたマルウェアの発生源を識別可能なボットネットトラッキングの3つの手法を実装し，評価検証することが挙げられる．

参 考 文 献

[1] P. Wang, S. Sparks, and C. Zou, “An advanced hybrid peer-to-peer botnet”, Dependable and Secure Computing, IEEE Transactions on, vol. 7, no. 2, pp. 113-127, 2010.

[2] サイバークリーンセンター（CCC）, https://www.ccc.go.jp/.

[3] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “A multifaceted approach to understanding the botnet phenomenon”, Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, IMC ’06, New York, NY, USA, pp. 41-52,ACM, 2006.

[4] N. Provos and T. Holz, “Virtual honeypots: from botnet tracking to intrusion detection”, ch. Tracking Botnets, Addison Wesley Professional, 2007.

[5] H. Zeidanloo and A. Manaf, “Botnet command and control mechanisms”, Computer and Electrical Engineering, ICCEE ’09. Second International Conference on, pp. 564-568, 2009.

[6] E. Hellweg, “When Bot Nets Attack” MIT Technology Review, September 24 2004.

[7] B. McCarty, “Botnets: big and bigger,” Security Privacy, IEEE, vol.1, no.4, pp.87–90, 2003.

[8] L. Spitzner, Honeypots: Tracking Hackers, Addison Wesley, September 13, 2002.

[9] 桑原和也，菊池浩明，寺田真敏，藤原将志，“ボットネットの連携感染を判定する発見的手法について”，情報処理学会論文誌，Vol. 51，No. 9，pp. 1600-1609，2010.

[10] 畑田充弘，中津留勇，秋山満昭，“マルウェア対策のための研究用データセット ～MWS 2011 Datasets～”，マルウェア対策研究人材育成ワークショップ 2011（MWS2011），pp. 1-5，2011.

[11] R. Agrawal, T. Imielinski, A. Swami，“Mining Association Rules between Sets of Items in Large Databases”, Proceedings of ACM SIGMOD-93, pp. 207-216, 1993.

[12] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.C. Hsu, “Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth”, Proc. 17th Int. Data Engineering Conf, pp. 215-224, 2001.

[13] R. Agrawal and R. Srikant, “Mining sequential patterns”, Data Engineering, International Conference on, vol.0, p. 3, 1995.

[14] F. Pedro, “A survey on sequence pattern mining algorithms.” University of Informatics, Gualtar, Portugal., January 18, 2011, (available at http://alfa.di.uminho.pt/ pedrogabriel/papers/SM_survey.pdf).

[15] 小堀智弘，菊池弘明，寺田真敏，“マルウェアの通信履歴と定点観測の相関について”，マルウェア対策研究人材育成ワークショップ 2008（MWS2008），2008.

[16] Christian Borgelt, “Apriori - Association Rule Induction“, http://www.borgelt.net/apriori.html.

[17] Trend Micro Threat encyclopedia, http://about-threats.trendmicro.com/.

[18] O. Thonnard and M. Dacier, “A framework for attack patterns’ discovery in honeynet data”, Digital Investigation, Vol. 5, No. Supplement 1, pp. S128-S139, 2008.

[19] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, “Statistical approaches to DDoS attack detection and response”, Proceedings of DARPA Information Survivability Conference and Exposition, vol.1, pp.303-314,2003.

[20] T. Nakashima, S. Oshima, Y. Nishikido, and T. Sueyoshi, “Extraction of characteristics of anomalously accessed IP packets by the entropy-based analysis”, Proc. Int. Conf. Complex, Intelligent and Software Intensive Systems CISIS 2008, pp.141-147, 2008.

[21] R. Lyda and J. Hamrock, “Using entropy analysis to find encrypted and packed malware”, IEEE Security & Privacy, vol.5, no.2, pp.40–45, 2007.

[22] Trend Micro Threat Encyclopedia, ”TSPY_KOLABC.CH Technical Details”, http://about-threats.trendmicro.com/ArchiveGrayware.aspx?language=en&name=TSPY_KOLABC.CH.

[23] L. Wei, T., Mahbod, and G. Ali A., “Automatic discovery of botnet communities on large-scale communication networks”, In ASIACCS ’09: Proc. the 4th Int. Symposium on Information, Computer, and Communications Security, pp. 1-10, 2009.

[24] Gu, G., Perdisci, R., Zhang, J. And Lee, W., “BotMiner: clustering analysis of network traffic for protocol and structure-independent botnet detection”, In 17th Usenix Security Symposium (2008), 2008.

[25] Husna, H., Phithakkitnukoon, S., Palla, S. and Dantu, R., “Behavior analysis of spam botnets”, Communication Systems Software and Middleware and Workshops, COMSWARE 2008. 3rd International Conference on, pp. 246-253, 2008.

業 績 リ ス ト

[1] 大類将之，菊池浩明，寺田真敏，“分散ハニーポット観測からのダウンロードサーバ間のアソシエーションルール抽出”，マルウェア対策研究人材育成ワークショップ 2009（MWS 2009），pp. 151-156，Oct. 2009.

[2] Nur Rohman Rosyid, Masayuki Ohrui, Hiroaki Kikuchi, Pitikhate Sooraksa and Masato Terada, “Frequent Sequential Attack Patterns of Malware in Botnets”, IPSJ SIG Technical Report, Vol. 2010-CSEC-48, No. 37, pp. 1-7, Mar. 2010.

[3] Masayuki Ohrui, Hiroaki Kikuchi and Masato Terada, “Mining Association Rules Consisting of Download Servers from Distributed Honeypot Observation”, The 13th International Conference on Network-Based Information Systems (NBiS 2010), pp. 541-545, Sep. 2010.

[4] Nur Rohman Rosyid, Masayuki Ohrui, Hiroaki Kikuchi, Pitikhate Sooraksa and Masato Terada, “A Discovery of Sequential Attack Patterns of Malware in Botnets”, The 2010 IEEE International Conference on Systems, Man and Cybernetics (IEEE SMC 2010), pp. 2564-2570, Oct. 2010.

[5] 大類将之，菊池浩明，寺田真敏，Nur Rohman Rosyid，“CCC DATAset におけるマルウェアの変遷”，マルウェア対策研究人材育成ワークショップ 2010 （MWS 2010），pp. 903-908，Oct. 2010.

[6] Masayuki Ohrui, Hiroaki Kikuchi, Masato Terada and Nur Rohman Rosyid, “Apriori-PrefixSpan Hybrid Approach for Automated Detection of Botnet Coordinated Attacks”, The 14th International Conference on Network-Based Information Systems (NBiS 2011), pp. 92-97, Sep. 2011.

[7] Nur Rohman Rosyid, Masayuki Ohrui, Hiroaki Kikuchi, Pitikhate Sooraksa and Masato Terada, “Analysis on the Sequential Behavior of Malware Attacks”, IEICE Transactions on Information and Systems, Vol. E94-D, No. 11, pp. 2139-2149, Nov. 2011.

謝 辞

本論文を執筆するにあたり，多くの方から多大なる御指導，御鞭撻を賜りました．

特に，研究に関わらず私を導いて下さった東海大学情報通信学部通信ネットワーク工学科 菊池 浩明 教授に深甚なる感謝を申し上げます．

また，本研究を推進するにあたって，懇切なる御教示並びに御激励を賜りました東海大学情報理工学部情報科学科 中西 祥八郎 教授，東海大学情報理工学部情報科学科 内田 理 准教授に厚く御礼申し上げます．

さらに，東海大学・中央大学・株式会社日立製作所による合同研究プロジェクト Scanners の一員として，活発な議論及び技術的な御助言，御示唆を賜わった株式会社日立製作所 寺田 真敏 氏，藤原 将志 氏，仲小路 博史 氏，鬼頭 哲郎 氏，東海大学 松尾 俊治 氏，桑原 和也 氏，中央大学 安藤 槙悟 氏，Scanners OB である小堀 智弘 氏に深く御礼申し上げます．

また，マルウェアに関する共同研究を行い，有益な意見を下さったキングモンクット工科大学ラカバン校の Nur Rohman Rosyid 氏に深く感謝致します．(I am grateful to Mr. Nur Rohman Rosyid at King Mongkut’s Institute of Technology Ladkrabang for his useful suggestions, who did joint research on the malware.)

そして，2 年間共に楽しみ，苦しみ，励まし合い，時には研究に対して有益な意見を与えてくれた東海大学大学院工学研究科情報理工学専攻の皆様，先生がたに感謝致します．

最後に，家族に心から感謝の意を表すると共に，謝辞とさせて頂きます．