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

Masayuki Ohrui
Hiroaki Kikuchi
Department of Information Science and Engineering,
Graduate School of Engineering, Tokai University
4-1-1 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan
yama@cs.dm.u-tokai.ac.jp
kikn@cs.dm.u-tokai.ac.jp

Masato Terada
Hitachi Incident Response Team (HIRT),
Hitachi, Ltd.
890 Kashimada, Kawasaki, Kanagawa, 212-8567, Japan
masato.terada.rd@hitachi.com

Nur Rohman Rosyid
King Mongkut's Institute of Technology Ladkraban
Chalongkrung Road, Ladkrabang Bangkok 10520, Thailand
nrohmanr@ugm.ac.id

***Abstract*—This paper aims to detect features of coordinated attacks by applying data mining techniques, Apriori and PrefixSpan, to the CCC DATAset 2008-2010 which consists of the captured packets data and the downloading logs. Data mining algorithms allow us to automate detecting characteristics from large amount of data, which the conventional heuristics could not apply. Apriori achives high recall but with false positive, while PrefixSpan has high precision but low recall. Hence, we propose hybriding these algorithms. Our analysis shows the change in behavior of malware over the past 3 years.**

I. Introduction

Malware has been improved in recent years. For example, many variants of malware are used for infection using multiple download servers controlled by some. This avoids researchers from tracing the source of malware developers. In particular, an advanced technique refereed as *the botnet coordinated attacks* with multiple servers makes detection of malwares to be extremely difficult.

Moreover, in recent years, "Gumblar" and other Web-based malware newly introduced an attack called *drive-by-download* which involved many web servers to make victim hosts downloading malware, resulting increase of the damage. It is almost impossible to manually trace the path of downloads because of the quantities and kinds of packets used to the drive-by-download attack. Instead, we need to use an algorithm of data mining for analysis.

There are two major data mining technique for extracting a valuable features of the malware from downloading logs — *Apriori* [1] and *PrefixSpan* [2]. The *Apriori* can be used to detect the association rule of the malware for coordinated attacks [3]. It was designed to detect significant correlations of set of items for extracting rules of items with high support (a fraction of the subset of items).

The support is useful feature for detecting all possible coordinated behaviors among servers. However, since *Apriori* deals with *subset of downloading events* without considering the order of events, it has high false positive ratio. For instance, a sequence of events a and then b is equivalent to one of b and a in *Apriori*. The detected coordinated patterns in *Apriori* contain some false coordinations that two independent servers happened to work at almost same time by chance. Hence, its confidence is not so high. While, *PrefixSpan* considers the *sequence of downloading events* that was ignored in *Apriori*. Hence, it is expected to have higher accuracy than *Apriori*. However, *PrefixSpan* does not evaluate the support of rule. Therefore, using sequential patterns mining in *PrefixSpan*, we can improve accuracy of the association rules considering time series of downloading events that was the drawback of *Apriori* [4]. Table I shows summary of comparison between *Apriori* and *PrefixSpan*.

In this paper, we examine two data mining techniques, *Apriori* and *PrefixSpan*, based on the dataset of actual downloading events, referred as CCC DATAset 2008-2010 [5], [6]. We focus our analysis on the change of behavior of malware over the past 3 years. Our experimental analysis, shows the investigated feature and changes in coordinated attacks. Interestingly, the number of malware infections has been decreasing for these 3 years. This suggests us that the main stream of botnet attack has been shifted from a single server to the coordinated servers with web-based drive-by-downloading malware.

II. Building Blocks

A. Apriori Algorithm

Apriori is a well-known algorithm for association rule discovery due to Agrawal et al. [1]. It allows to efficiently discover useful association rules by excluding the rules

Table I
THE DIFFERENCE BETWEEN APRIORI AND PREFIXSPAN

	Apriori	PrefixSpan
Proponent	Agrawal, et al.[1]	Pei, et al.[2]
Extraction	Association rule (A, B → C)	Sequencial pattern (A, B, *, C)
Precision	Support, Confidence	Confidence
Feature	A set of items (unordered)	Sequence (in order)

Table II
EXAMPLE OF TRANSACTION

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

Table III
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`

those support and confidence smaller than giving minimum support and confidence. With the minimum support, we can squeeze many useless rules to be examined.

Association rules is of the form

$$X(antecedent) \Rightarrow Y(consequent)$$

from a given set.

A *support* is a probability of set of an association rule ($X \Rightarrow Y$) to be shown out of all transactions N, which is defined as

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

A *confidence* is a probability of the rule is satisfied, namely, a chance of Y is true if X is true. The definition is given by

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

For instance, the association rule $B, C \Rightarrow E$ in Table II has support and confidence as

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

$$\mathit{Conf}(B, C \Rightarrow E) = 2/2 = 1.$$

Thus, the rule of $B, C \Rightarrow E$ is support 50% and confidence 100%. In other word, this rule shows with probability of 50%, and $B, C \Rightarrow E$ appears with probability of 100% when B and C appear.

B. PrefixSpan Algorithm

Sequential pattern mining is a method to discover subsequence patterns in database of sequences, where each sequence consists of a list of elements and each element consists of a set of items. Given a user-specified minimum support threshold as a condition, sequential pattern mining is to find all of the frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is greater than or equal to the minimum support. Sequential pattern mining method, called *PrefixSpan* (i.e., Prefix-projected Sequential pattern mining) was firstly proposed by Jien Pei [2].

Let a_i, b_j be items; α_i, β_j be sequences of item; $\alpha = \langle a_1 a_2 ... a_n \rangle$ and $\beta = \langle b_1 b_2 ... b_m \rangle$. Then α is **subsequence** of β, denoted by $\alpha \sqsubseteq \beta$ if and only if, there exist integers $j_1, j_2, ..., j_n$ such that $1 \leq j_1 < j_2 < ... < j_n \leq m$, such that $a_1 = b_{j_1}, a_2 = b_{j_2}, ..., a_n = b_{j_n}$. A **sequence database** S is a set of tuples $\langle sid, s \rangle$, where sid is a **sequence_id** and s is a **sequence**. The support of a sequence α in a database S is the number of tuples in the database containing α, i.e., $support(\alpha) = |\{\langle sid, s \rangle | \langle sid, s \rangle \in S, \alpha \sqsubseteq s\}|$. Given a positive integer *min_sup* as a support threshold, a sequence α is called a **frequent sequential pattern** in database S if the sequence is contained by at least *min_sup* tuples in the database, i.e., $support(\alpha) \geq$ *min_sup*. The number of item in a sequence is called the **length** of the sequence, so, sequential pattern with length ℓ is called **ℓ-pattern**.

In term of *PrefixSpan* algorithm; Let α and β be sequences $\langle a_1 ... a_n \rangle$ and $\langle b_1 ... b_m \rangle$, respectively.

1) **Prefix** and **Postfix** : sequence α is prefix of β if and only if, $a_i = b_i$ for $i = 1, ..., m$. For example, $\langle a\ a\ b\ c \rangle$ is prefix of $\langle a\ a\ b\ c\ d\ d\ a\ b \rangle$ and sequence after prefix is postfix, $\langle d\ d\ a\ b \rangle$ is postfix in $\langle a\ a\ b\ c\ \mathbf{d\ d\ a\ b} \rangle$.
2) **Projection** : Let α, β, γ be sequences such that $\beta \sqsubseteq \alpha, \gamma \sqsubseteq \alpha$. Sequence γ is β-**projection** of α if and only if (1) β is prefix of γ, and (2) there exists no longer subsequence of α such that β is its prefix. For example, **c**-projection of $\langle a\ a\ b\ \mathbf{c}\ d\ c\ d\ a\ b \rangle$ is $\langle d\ c\ d\ a\ b \rangle$.

(Example 1) Given a sequence database S in Table III and user specified $min\_sup = 2$, sequential patterns in S can be mined by *PrefixSpan* method in the following steps:

Step 1: Find 1-pattern sequence.

Scan database S once to discover all frequent items in sequences. These are ⟨`PE`⟩ :5, ⟨`WO`⟩:5, ⟨`TR`⟩:5, ⟨`BK`⟩:2 and ⟨`TS`⟩:2, where ⟨`pattern`⟩:$count$ is a pair of the pattern and support count.

Step 2: Distribute search space.
The projected database can be distributed into the following five subsets according to the five prefixes which resulted from step 1: (1) the ones having prefix ⟨`PE`⟩;...; and (5) the ones having prefix ⟨`TS`⟩.

Step 3: Find subsets of sequential patterns.
These can be mined by constructing corresponding *projected databases* and delved each recursively.

III. The botnet coordinated attacks

A. Definition

The botnet has a feature that coordinated attacks of multiple servers making a victim infected by a set of malwares [7]. For example, Table IV shows sequential infections observed the Cyber Clean Center (CCC) DATAset 2009, the captured packets data by 94 honeypots [5] in which a host is infected by three malwares, `PE_VIRUT.AV`, `TROJ_BUZUS.AGB` and `WORM_SWTYMLAI.CD` as scheduled in the same way. Although the these servers are assigned different IP addresses, it turns out to be a correlation in the malware infections. In this paper, we call the multiple infections made by several servers *the botnets coordinated attacks*.

B. Experimental Data

In order to verify efficiency of our proposed method, we apply Apriori and PrefixSpan algorithm to the experimental data, CCC DATAset 2008–2010. The 94 independent honeypots have observed malware traffic at the Japanese tier-1 backbone under coordination of the CCC. CCC DATAset consists of the access log of attack for 3 years during November 1, 2007 until April 30, 2010. The honeypots are periodically rebooted every 20 minutes. We call the time interval *time slot* throughout this paper. Observation in a day gives 72 time slots. A transaction consists of malware names that are downloaded in a time slot. Similarly, malware downloading logs is divided in terms of time slots. In out experiment, we use vulnerable Windows XP as honeypot.

IV. Hybrid Approach of Apriori and PrefixSpan

A. Comparison between Apriori and PrefixSpan

We evaluate two automated algorithms, Apriori and PrefixSpan, in terms of accuracy in detecting malware coordinated attacks. Table V shows a part of experimental result in few day early 2009. Our target coordinated attack to be detected in these algorithms is of sequence of malware, `TSPY_KOLABC.CH`, `WORM_SWTYMLAI.CD` and `BKDR_POEBOT.GN` that was reported by the Trend Micro [8]. The accuracy of Apriori is given as a frequency of detected time slots, indicated in columns labeled as “Slots” out of true time slots defined by manual investigation, while the accuracy of PrefixSpan is defined as fraction of detected coordinated attack patterns out of true patterns, labeled as “Ptns” in the table.

Table VI
Accuracy in Apriori

	Coordination	Non-Coordination	Sum
Extracted	315	149	464
Non-Extracted	0	N/A	N/A
Sum	315	149	464

Table VII
Accuracy in PrefixSpan

	Coordination	Non-Coordination	Sum
Extracted	482	0	482
Non-Extracted	93	N/A	93
Sum	575	N/A	575

For example, Apriori surely extracts all four coordinated attacks in 3rd February. The Prefix spans detects three correct patterns, missing 6 patterns out of 9, in the same day. In 28th February, Apriori has false detections for 7 slots. The reason of false positive is that Apriori considers all possible combinations of malware without seeing the order of detection. On the other hand, PrefixSpan has relatively low false positive than Apriori, though it implies high false negative. For instance, in 4th February, it has $29-(3+7+4+12) = 3$ missing patterns with too low frequency.

Consequently, Apriori is good at detecting possible time slot when coordinated attacks may have, while PrefixSpan is useful for detecting exact coordinated patterns of malware. We can combine these two automated approach for accurate detection of attacks.

B. Accuracy in Detection

Our comprehensive investigation of CCC DATAset is summarized in Table VI and VII, accuracy of Apriori and PrefixSpan, respectively. Note that Apriori aims to detect coordinated time slots and PrefixSpan detects sequence patterns of malware. Table shows that Apriori has 149 false positive (slots) out of 464 and no false negative, and PrefixSpan has no false positive (patterns) but fails to detect 93 patterns out of 575. In summary, we show two criteria, *precision*, defined as a fraction of correctly detected slots (patterns) in all detected slots and *recall*, defined as a fraction of correctly detected slots (patterns) in all slots with attacks in Table VIII. Apriori archives high recall but with false positive. PrefixSpan can be tuned with appropriate minimum support bound to filter out useless patterns.

C. Hybrid Approach of Apriori and PrefixSpan

From our observation, we come up with idea of hybriding Apriori and PrefixSpan. Firstly, we apply Appriori to detect potential time slots with coordinated attacks since we have no Apriori knowledge that which malware are likely to be

Table IV
SAMPLE OF COORDINATED ATTACKS OF MALWARE

Time	Source IP address	Dst Port	Protocol	MW
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

Table V
COMPARISON BETWEEN APRIORI AND PREFIXSPAN

Date	Apriori			PrefixSpan		
	Rule	Slots	True [Slots]	Rule	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	
Sum		464	315		482	575

Table VIII
RECALL AND PRECISION

	Apriori	PrefixSpan
Recall	$315/315 = 1$	$482/575 = 0.838$
Precision	$315/464 = 0.678$	$482/482 = 1$

correlated to others. After Apriori filtered out possible slots, we apply PrefixSpan algorithms to improve accuracy. For example, in February 4th, each of Apriori and PrefixSpan detects 9 patterns and 32 patterns, respectively. However, after Apriori detects three major malwre, TSPY, WORM, and BKDR, the second filter of PrefixSpan reduces the number of false alerts from 32 to 4 patterns, listed in Table V labeled as “PrefixSpan”. The results suggests that fourth patterns are the most likely sequences of malware used in a botnet. In simplicity, we concentrate three interested malware in this example. In practical, we should deal with many unrelated malware observed in the same period of time.

V. EXPERIMENTAL RESULT

A. Change in Malwares

We investigated the downloading logs for 3 years in terms of change of malware. Table IX shows the common malware being detected for 3 years. We note that PE_VIRUT.AV is a high-ranked malware for 3 years and the PE_VIRUT.AV is the malware that begins of the coordinated attacks. Also, PE is the most common malware family name, though the number of infections is decreasing.

Next, we focused on the IRC servers and the DNS servers used for the coordinated attacks shown in Table X and Table XI, respectively. Table shows unique IP addresses of servers for each slot. The most common IRC server for 3 years is hub.*****.com. The IRC domain was used when coordinated attacks begin with PE. Similarly, some DNS domains have been used for 3 years (indicated as bold in Table XI). Therefore, we conclude that the coordinated attacks has been attempted for 3 years long.

B. Change in Coordinated Attacks

Firstly, Figure 1 shows observed numbers of coordinated attacks for 2 years. We use Apriori for computing the monthly average frequencies of association rules in all honeypots of 730 days. For reason of reliable analysis, we exclude the fault of identification of malware, labeled as “UNKNOWN”. The extracted rule is composed of more than three kinds of malware events, From Figure 1, we observe that the number of coordinated attacks is decreasing similarly as the number of malwares decreases, too. Our analysis of the captured packets data reveals the decrease of diversity of attacks. For example, The coordinated attacks were made in three different patterns in 2009 , but a single pattern is attempted in 2010.

Secondly, we investigate how many kinds of the malware is used to perform coordinated attacks. For this purpose, we

Table IX
Common Malware Names Observed in 2008-2010

MW	2008		2009		2010	
	Rank	Uniq.	Rank	Uniq.	Rank	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

Table X
Servers Observed in 3 Years

Rank	2008		2009		2010	
	IRC Domain	Num.	IRC Doamin	Num.	IRC Domain	Num.
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

Table XI
Domains Used for Attacks

Rank	DNS Domain	Num.	2008	2009
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	1
10	**ss.nadnad***.info**	16	81	5
11	**ss.MEMEH***.INFO**	15	90	
12	videogale***.com	12		
13	blah.swapixtr***.com	10		
26	xx.nadna***.info	2		

applied PrefixSpan algorithm because it can distinguish the patterns with different infection order, hopefully extract the coordinated infection patterns of all honeypots.

Figure 2 shows the change in the average number of kinds of malware used to attack. The number of kinds of malware increases, contrary to the decrease in the overall attacks in total. We stress that this shows the coordinated attacks are getting more complex and advanced than before. As a result, the malware downloaded with HTTP GET which was used by two malware in 2008 and 2009, but was observed five times in malware in 2010. Thus, we conclude that the coordinated attacks is obviously complicated.

Finally, we investigate the lifecycle of the coordinated attacks. Figure 3 shows the distribution of active durations of coordinated attacks non-duplicate with three malwares. In general, the lifecycle of coordinated attacks is very short for 1 month from 2 weeks. For example, the malware that cooperates with PE_VIRUT.AV is changing every year.

C. Consideration

The reason why number of coordinated attacks is decreasing is that the number of downloads is decreasing as shown in Figure 4. We claim that it is an evidence that major attack method is replaced by web-based one in 3 years.

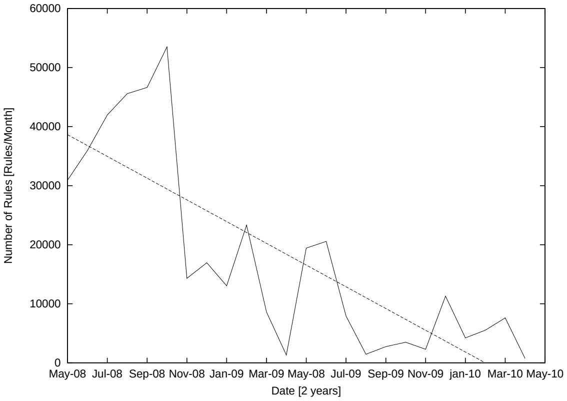

Figure 1. Number of Coordinated Attacks (2009-2010)

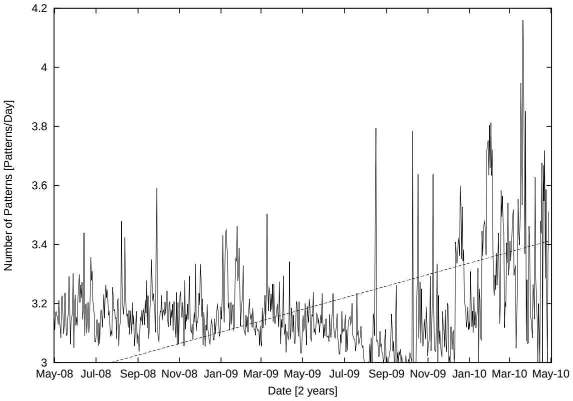

Figure 2. Average Langth of Coordinations

VI. Conclusions

We have reported the characteristics and evolution of the coordinated attacks using the CCC DATAset for the past 3 years. While the number of coordinated attacks decreased, the number of distinct malware that used for the coordinated attacks has been increased.

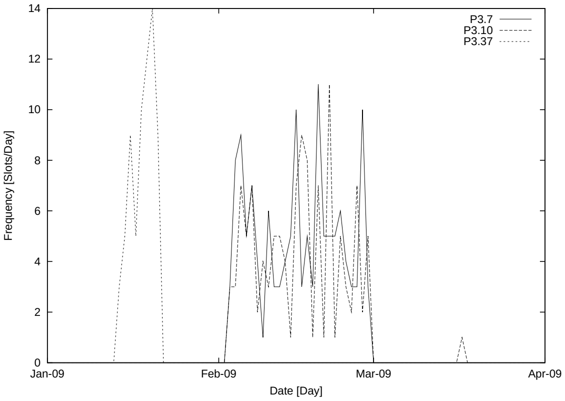

Figure 3. Duration of Coordinated Attacks [4]

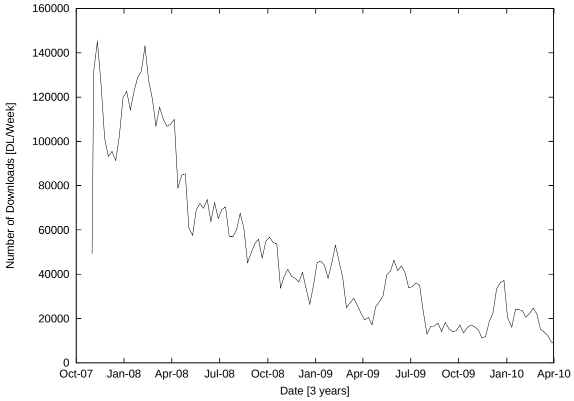

Figure 4. Number of Downloads (2007-2010)

Acknowledgement

We thank Mr. Masashi Fujiwara at Hitachi Ltd. for their useful suggestions.

References

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

[2] J. Pei, J. Han, MA Behzad, and H. Pinto, “PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth”, Proc. of the 17th Int’l Conf. on Data Engineering, pp. 215-224, 2001.

[3] M. Ohrui, H. Kikuchiand M, Terada, “Mining Association Rules Consisting of Download Servers from Distributed Honeypot Observation”, The 13th Int’l Conf. on Network-Based Information Systems (NBiS 2010), pp. 541-545, 2010.

[4] N. R. Rosyid, M. Ohrui, H. Kikuchi and P. Sooraksa, M. Terada, “A Discovery of Sequential Attack Patterns of Malware in Botnets”, The 2010 IEEE Int’l Conf. on Systems, Man, and Cybernetics (SMC 2010), pp. 2564-2570, 2010.

[5] M. Hatada, Y. Nakatsuru, M. Terada and Y. Shinoda, “Dataset for Anti-Malware Research and Research Achievements Shared at the Workshop”, IPSJ Malware Workshop 2009 (MWS 2009), pp. 1-8, 2009 (in Japanese).

[6] M. Hatada, Y. Nakatsuru, M. Terada and Y. Shinoda, “Datasets for Anti-Malware Research MWS 2010 Datasets”, IPSJ Malware Workshop 2010 (MWS 2010), pp. 1-5, 2010 (in Japanese).

[7] K. Kuwabara, H. Kikuchi, M. Terada and M. Fujiwara, “Heuristics for Detecting Botnet Coordinated Attacks”, The 4th Int’l Workshop on Advances in Information Security (WAIS 2010), pp. 603-607, 2010.

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