A Discovery of Sequential Attack Patterns of Malware in Botnets

Nur Rohman Rosyid*, Masayuki Ohrui*, Hiroaki Kikuchi*, Pitikhate Sooraksa† and Masato Terada‡
*School of Science and Technology
Tokai University, 1117 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan
Email: nrohmanr@ugm.ac.id; kikn@cs.dm.u-tokai.ac.jp
†King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd., Bangkok, Thailand
Email: kspitikh@kmitl.ac.th
‡Hitachi Incident Response Team (HIRT), Hitachi, Ltd.
890 Kashimada, Kawasaki, Kanagawa, 212-8567, Japan
Email: masato.terada.rd@hitachi.com

***Abstract*—More than 90 independent honeypots have observed malware traffic at the Japanese tier-1 backbone. Typical attacks were made by multiple servers, coordinating to send many kinds of malware. This paper aims to discover some frequent new sequential attack patterns of malware. It is not easy to identify particular patterns logs of one year because the volume of dataset is too large to investigate one by one. To overcome the problem, this paper proposes data mining algorithm, the *PrefixSpan* method. We implement the *PrefixSpan* algorithm to analyze the malware footprints and show the experimental result. The result of analysis shows that the attacks are performed by multiple sequential attack patterns within a short amount of time.**

***Index Terms*—PrefixSpan, Malware, Botnets, Coordinated Attack, Sequential Pattern.**

I. Introduction

A botnet's attack goes very active by sending malware to infect the targets on the Internet. Previously, the conventional attackers who aim to attack targets use the integrated tools including buffer overflow, port-scan, trojan horse, worm, etc. The threats of these techniques are quite easy to be anticipated by antivirus software based on the signature file of malware. Currently, the improved method splits the tool into small parts of specific functions such as malware, trojan horse, worm, etc.; these are distributed toward the downloading servers (DS) through the Internet. The DS is a host that have been compromised by malware, afterward, the attacker sends control messages to the target of attack from the Command and Control (C&C) server. The attacker manipulates and reconfigures the attack according to their needs. The attacker usually utilize the Internet Relay Chat (IRC) server to send commands to DSs. This is how a botnet system works. The attacks are coordinated systematically under the botnet's attack strategy. In this paper, we call the sequential attacks by botnets *the coordinated attack*.

The conventional antivirus software based on the signature of single malware can not easily detect the variety of the coordinated attack. Fortunately, Botnet's activity can be traced by the observation of malwares' footprints on several DSs that spreads on the network; this method is a part of honeypot system. The honeypot is a decoy host, pretending to be a vulnerable computer and it looks attractive toward the attackers. Honeypot reboots every 20 minutes, and during this time, honeypot is recording every inbound packet as an access log consists of Timestamp, Honeypot ID, Source/Destination port number, Source IP address, Source port number, Hash value(SHA1), Malware name, and Malware file name. This 20 minutes duration is called a time slot, or simply slot.

More than 90 independent honeypots have observed malware traffic at the Japanese tier-1 backbone under coordination of the Cyber Clean Center (CCC). CCC DATAset 2009 consists of the access log of attack for a year during May 1, 2008 until April 30, 2009. This paper explores and discovers the coordinated attack patterns in the CCC DATAset 2009. Since botnet utilizes systematic attack method, the sequence of malware have been downloaded by honeypots must be in a particular form of coordinated pattern.

Heuristic techniques for detection of malware have made by botnet's coordinated attack [1], it gives useful information for determination of the characteristics and relationship of botnet's attack. The confidence of occurrence in association rules of malware attacks based on Apriori algorithm [2], contributes valuable knowledge in the study of CCC DATAset 2009. Exploration of another dataset such as clustering attack pattern with an appropriate similarity measure on time series analysis, enables the identification of several malwares' activities in the traffic have collected by the honeypots [3]. In addition, implementation of hybrid method, data mining and expert system can develop a knowledge base of the malware behavior patterns [4].

Our contribution of this work is to propose a new method to detect coordinated of several servers with high accuracy as the frequent sequential attack pattern. Our proposed method is based on a data mining algorithm, *PrefixSpan* algorithm due to Pei at al [7]. The *PrefixSpan* method [7] is an algorithm for efficient mining of sequential pattern in a huge dataset without the requirement to construct candidate generation and the memory consumption is much smaller [8] and hence

978-1-4244-6587-3/10/$25.00 É2010 IEEE

TABLE I
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		

PrefixSpan algorithm is applied in this research.

The rest of the paper is organized as follow. Section II introduces the basic concept of *PrefixSpan* algorithm. Section III shows our framework for mining sequential attack pattern of malware. Section IV shows the relation of the attack pattern and Source IP address and timestamp. Section V concludes this paper.

II. MINING THE SEQUENTIAL PATTERNS

Sequential pattern mining is a method to discover subsequence patterns in database. This study was introduced by Agrawal R. [5] and this concept is described as follow: *Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and 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 [7], which discovers frequent subsequences as patterns in a sequence database.

TABLE II
SEQUENTIAL PATTERN

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

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 I 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.

Starting from prefix ⟨PE⟩, we can make ⟨PE⟩-projected database that consists of five postfix sequences: ⟨WO TR⟩, ⟨TR WO⟩, ⟨TR TS WO⟩, ⟨PE TR WO BK⟩, and ⟨WO TR WO⟩. Recursively, back to the step 1 by scanning ⟨PE⟩-projected database once, all 2-pattern sequences having prefix ⟨PE⟩ can be found, these are: ⟨PE WO⟩:5 and ⟨PE TR⟩:5. Then ⟨PE⟩-projected database is divided into two subsets according to the two prefixes, *i.e.*, ⟨PE WO⟩ and ⟨PE TR⟩. Afterward, each generated projected database is mined recursively. From prefix ⟨PE WO⟩ having three postfix sequences ⟨TR⟩, ⟨BK⟩, and ⟨TR WO⟩, mining these sequences results sequential pattern ⟨PE WO TR⟩, which can not be scanned anymore because its support count is less than min_sup. From prefix ⟨PE TR⟩ having four postfix sequences ⟨WO⟩, ⟨TS WO⟩, ⟨WO BK⟩, ⟨WO⟩, we have resulting 3-pattern ⟨PE TR WO⟩:4. The final projected database as well as sequential patterns are listed in Table II.

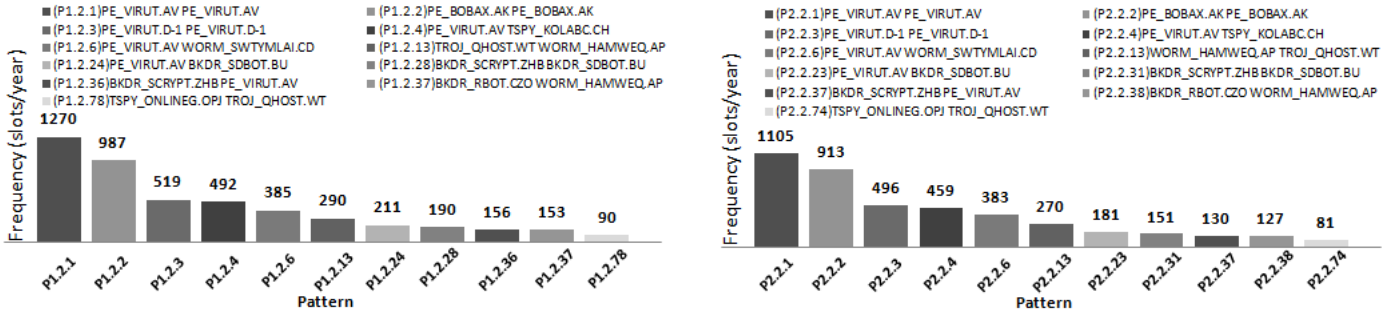
III. MINING SEQUENTIAL PATTERN OF MALWARE

A. Input Data

We explore the CCC DATAset 2009 to discover frequent attack patterns based on the sequence of malware have been downloaded by the honeypots. In this experiment, we investigate a year-long access log was recorded by two of the

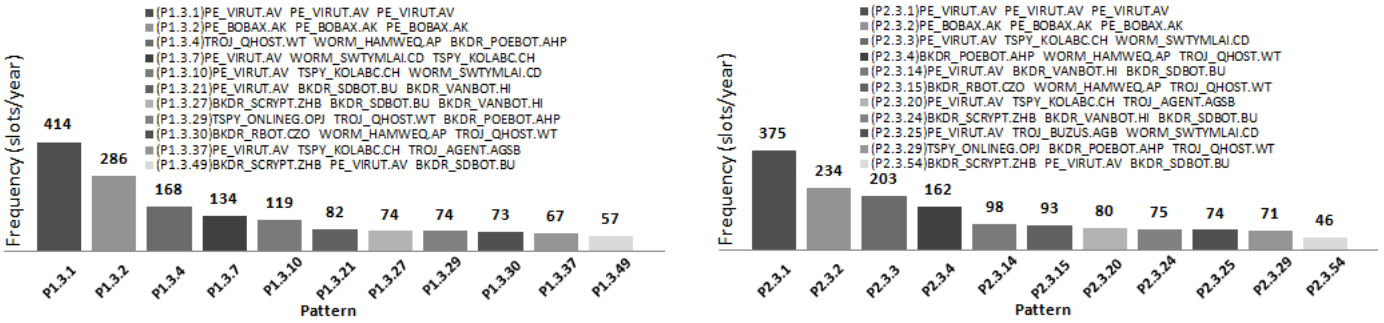
TABLE III
SAMPLE OF PRE-PROCESSING DATA (SEQUENCE DATABASE)

Slot	Sequence of Malware
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



(a) Honeypot 1 (b) Honeypot 2

Fig. 1. Sequential attack 2-pattern of malware



(a) Honeypot 1 (b) Honeypot 2

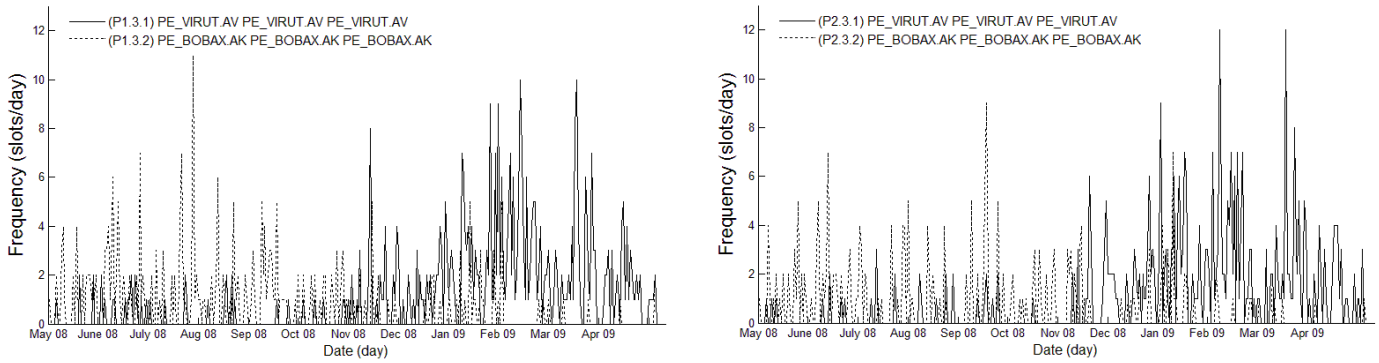
Fig. 2. Sequential attack 3-pattern of malware

honeypots out of 94 honeypots. Both honeypots run different operating system; honeypot number 1 runs under Windows XP+SP1 and number 2 runs under Windows 2000. For this purpose, we perform pre-processing access log, so it is compatible with *PrefixSpan* algorithm. An input is a text file consisting of *lines*, where each line is a sequence of malware names. The term *lines* are interchangeable with *slots* to satisfy the honeypot system's term in the discussion hereafter. The order of a sequence is determined by the malwares' download timestamps in one time slot. The average number of lines per honeypot through a year is 14.684 lines. The sample of the pre-processing data can be seen on Table III.

B. n-Pattern of Malware Attack

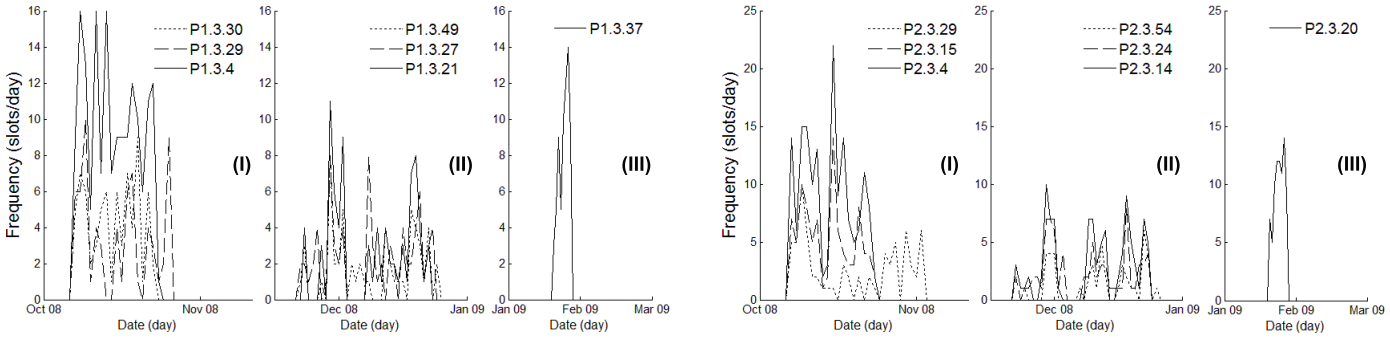
A purpose of this experiment is to provide valuable information about coordinated attack of the botnet system. Mining of CCC DATAset2009 using *PrefixSpan* algorithm results in the form of sequential attack pattern of malware that have attacked the honeypots. The sequential attack pattern of malware is displayed in the form of n-Pattern; where as mentioned before, n is the sequence length or the number of malware that composes a pattern. With n is 2, the result is a list of sequential attack 2-patterns of malware that satisfies the minimal support. The minimal support is defined as the number of slots have been infected by these patterns; this is also called download frequency. We are interested in both the download frequency and the download time period of botnet's attack during the year. Here, we investigate sequential attack 2-pattern and 3-pattern. So, this information can be consider as an early warning signal from the attack of the botnet system, as explained hereafter.

Figure 1 (a) and (b) shows the column charts of sequential attack 2-pattern for each honeypot number 1 and 2, respectively; the x-axis is a pattern name and y-axis is a down-



(a) Honeypot 1 (b) Honeypot 2

Fig. 3. Distribution of duplicate the sequential attack 3-patterns of malware within a year



(a) Honeypot 1 (b) Honeypot 2

Fig. 4. Distribution of non-duplicate the sequential attack 3-patterns of malware within a year

load frequency (*slots/year*). The minimum support threshold *min_sup* is 70 and maximum threshold of the sequence length *max_pat* is 2, this parameters are determined to discover sequential attack 2-pattern. The average number of slots in the honeypot per day is 72 *slots*; if any sequential attack 2-pattern attacks during the certain of a day intently and the download frequency is counted equal to or greater than 70 *slots*, then put it into account. Therefore, 70 is a reasonable as *min_sup*.

The sequential attack patterns are indexed into the form of $P_{h.x.y}$ to simplify naming pattern, where h is a honeypot number, x is a pattern length and y is a serial number in the list. For example, $P_{1.2.1}$ is a sequential attack 2-pattern with serial number 1 at honeypot number 1.

We classify mining result by the form of the malware's sequence that composes the pattern into two categories: *duplicate* and *non-duplicate*. The term duplicate is due to presence of more than once the same malware in the sequential pattern. For example, as shown in Fig. 1 (a), patterns $P_{1.2.1}$ and $P_{1.2.2}$; Fig. 2 (a), patterns $P_{1.3.1}$ and $P_{1.3.2}$; each of pattern is composed by the same malware such as `PE_VIRUT.AV` and `PE_BOBAX.AK`. The other category is called non-duplicate pattern.

This study has shown some significant relationship between both honeypots in terms of the order in the list of the sequential attack pattern and the download frequency in a year. The sequential attack 2-patterns of malware resides at the top five ranked in the list from both honeypots have same sequential pattern. As shown in Fig. 1 (a) and (b), the other sequential attack 2-patterns of malware only have a bit difference in the series number; only pattern $P_{1.2.13}$ and $P_{2.2.13}$ have a difference in sequential pattern. The difference in download frequency for each sequential attack pattern is relatively small. For example, the difference in download frequency between pattern $P_{1.2.6}$ (385 *slots/year*) and $P_{2.2.6}$ (383 *slots/year*) is 2 *slots/year*.

The value of parameter n influences the quantity of mining result either in the number of sequential attack pattern or in download frequency. For mining 3-pattern, We reduces the value of *min_sup* to 30 (40% out of 2-pattern *min_sup*) and n is equal to 3. Mining both honeypots extracts 169 and 118 of patterns that includes 29% and 26% non-duplicate pattern, respectively. Figure 2 (a) and (b) shows the column charts of sequential attack 3-pattern of malware for both honeypots.

The top ranked of the lists both sequential attack 2-pattern and 3-pattern are dominated by duplicate patterns consisting of `PE_VIRUT.AV` and `PE_BOBAX.AK`. This duplication implies that malware successfully infects the honeypot more than once in one slot. This facts may be regarded as indication that both malware are the most common malware have utilized by botnet system.

Figure 3 shows the distributions of duplicate sequential attack 3-patterns thet were the most frequently downloaded by

TABLE IV
LIST OF THE SEQUENTIAL ATTACK 3-PATTERN OF MALWARE

ID	Freq.	Sequential Attack Patterns	Ave[s]	SD[s]	Unique Host			Type	Gr.
$P_{1.3.4}$	168	TROJ_QHOST.WT WORM_HAMWEQ.AP BKDR_POEBOT.AHP	4.27	51.07	1	1	1	A_1E_1	A
$P_{1.3.29}$	74	TSPY_ONLINEG.OPJ TROJ_QHOST.WT BKDR_POEBOT.AHP	97.04	165.46	41	1	1	$A_4E_{1,3}$	A
$P_{1.3.30}$	73	BKDR_RBOT.CZO WORM_HAMWEQ.AP TROJ_QHOST.WT	56.65	235.71	3	1	1	A_1E_1	A
$P_{2.3.4}$	162	BKDR_POEBOT.AHP WORM_HAMWEQ.AP TROJ_QHOST.WT	34.12	175.92	8	1	1	A_1E_1	A
$P_{2.3.29}$	93	TSPY_ONLINEG.OPJ BKDR_POEBOT.AHP TROJ_QHOST.WT	72.66	191.33	34	1	1	A_4E_3	A
$P_{2.3.15}$	71	BKDR_RBOT.CZO WORM_HAMWEQ.AP TROJ_QHOST.WT	381.48	478.60	5	1	1	$A_1E_{1,3}$	A
$P_{1.3.21}$	82	PE_VIRUT.AV BKDR_SDBOT.BU BKDR_VANBOT.HI	108.31	212.90	48	1	1	$A_3E_{1,3}$	B
$P_{1.3.27}$	74	BKDR_SCRYPT.ZHB BKDR_SDBOT.BU BKDR_VANBOT.HI	732.12	422.57	11	1	1	$A_{3,5}E_3$	B
$P_{1.3.49}$	57	BKDR_SCRYPT.ZHB PE_VIRUT.AV BKDR_SDBOT.BU	862.60	304.87	5	42	1	$A_5E_{3,4}$	B
$P_{2.3.14}$	98	PE_VIRUT.AV BKDR_VANBOT.HI BKDR_SDBOT.BU	75.54	177.64	55	1	1	A_5E_3	B
$P_{2.3.24}$	75	BKDR_SCRYPT.ZHB BKDR_VANBOT.HI BKDR_SDBOT.BU	821.86	326.30	6	2	1	$A_{2,5}E_3$	B
$P_{2.3.54}$	46	BKDR_SCRYPT.ZHB PE_VIRUT.AV BKDR_SDBOT.BU	968.42	258.12	6	34	1	$A_5E_{3,4}$	B
$P_{1.3.37}$	67	PE_VIRUT.AV TSPY_KOLABC.CH TROJ_AGENT.AGSB	163.43	200.34	45	42	4	A_5E_3	C
$P_{2.3.20}$	80	PE_VIRUT.AV TSPY_KOLABC.CH TROJ_AGENT.AGSB	85.97	153.90	45	1	4	A_5E_3	C

both honeypots; the x-exis is the date in *day*, in one year; the y-exis indicates download frequency in *slots/day*. Each duplicate pattern is comprised of the malware PE_VIRUT.AV and PE_BOBAX.AK. The patterns of these attacks are distributed uniformly over a year. As shown in Fig. 3(a), pattern $P_{1.3.1}$ has two peaks on February and March 2009 with 10 *slots/day*, whereas pattern $P_{1.3.2}$ has observed at the maximum rate of 11 *slots/day* on August 2008. Similarly, Fig. 3(b) shows pattern $P_{2.3.1}$ has peaks at similar time-points as pattern $P_{1.3.1}$, but it has an infection rate of 12 *slots/day*. Pattern $P_{2.3.2}$ has a peak on September 2008 with 9 *slots/day*. These two patterns in both honeypots are composed by malware that have the ability to disable some services on systems running Windows 2000 and XP such as *Internet Connection Firewall* (ICF) and *Internet Connection Sharing* (ICS); they listen to varying ports and connects to an IRC server [9]. Regarding to the botnet's attack, our conjecture is the distribution charts as shown in Fig.3 can be considered as a distribution form of command and control (C&C) of botnet system.

The non-duplicate sequential attack 3-patterns have been observed by both honeypots, having similarity either the names of malware or the download time period. These top 60 ranked patterns on the list are classified into some groups by similarity of the download time period. This experiment found three groups that are interesting to investigate. Table IV shows the classification. Group A at both honeypots is composed by the five same malware, but there are some differences in the sequence of malware composes the pattern. Patterns $P_{1.3.30}$ and $P_{1.3.15}$, as shown in Table IV have exactly same sequential pattern, but have a bit difference in download frequency; it is 2 *slots/year*. Moreover, two sequential attack 3-patterns from both honeypots have the same series number in the list, and they are ranked as 4th and 29th; but the sequence of malware are reversed order.

Similarly, group B as shown in Table IV at both honeypots is composed by the four identical malware, but the sequence of malware is partially ordered. Both honeypots have one pattern: $P_{1.3.49}$ and $P_{2.3.54}$, respectively; these 3-patterns have entirely same sequential pattern, and they have difference in download frequency 11 *slots/year*.

The other group of sequential attack 3-patterns as presented in table IV is group C. This group consists of patterns $P_{1.3.37}$ and $P_{2.3.20}$; they are precisely identical either the malware that makes up or the sequence of malware.

Figure 4 shows the attacks' distributions of non-duplicate sequential attack 3-patterns during the year, which were classified into three groups: A, B, and C as mentioned before. Both honeypots were downloading group A throughout 20 days on October 2008, and the maximum infection rate of 16 *slots/day* and 22 *slots/day*, respectively; as shown in Fig.4(a-I) and (b-I). The activity of botnets' attacks from group B seems like sustained bursts (see [3]) throughout 25 days; started on November until December 2008, and the maximum infection rate of 11 *slots/day* that is occurred at honeypot number 1, as indicated on Fig.4(a-II) and (b-II). Patterns $P_{1.3.37}$ and $P_{2.3.20}$ are classified as a group C, and were downloaded in a short time period on last February 2009 throughout 8 days; these behavior also seems like sustained bursts and gain 14 *slots/day* as a maximum infection rate, as shown in Fig.4(a-III) and (b-III).

Both honeypots run different operating system. Our investigations found high similarities, but also some differences. So, we consider these differences are caused by operating system dependency. The common features of non-duplicate sequential attack 3-pattern are (1) occured intensively within the short time period (a narrow time period) during one month, (2) the number of slots that have been infected is greater than that of duplicate pattern, and (3) each group, except group C has been performed by multiple patterns.

In this experiment, We investigate time interval of the sequential attack 3-patterns downloaded into the honeypots. The time interval is defined by a time difference between the first and last malware downloads within the one sequential attack 3-pattern. We show each the average time interval of 3-pattern and its standard deviation, in Table IV. The distributions of each average time interval varies strongly. It may be outcome of the dynamic behavior of the Internet traffic causes the gap of time interval for a few download event or may be caused by multiple botnets' attacks. It is difficult to generalize a proper reason for this fact. In case of 3-patterns $P_{1.3.30}$ and $P_{2.3.15}$, each pattern has an average time interval less then 7 *minutes* and a big standard deviation greater than 7

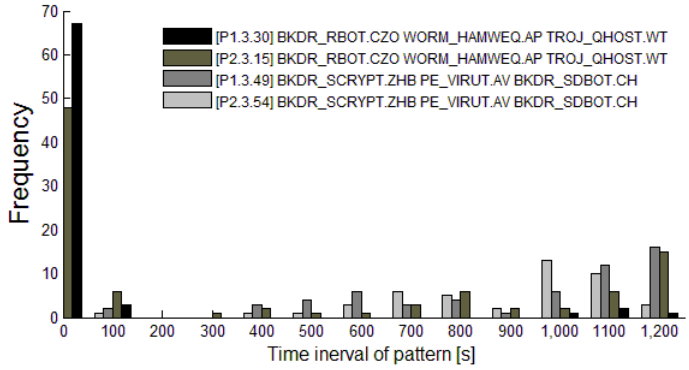


Fig. 5. Histogram of time interval of the sequential attack 3-pattern of malware

minutes, but the histogram of these distributions time interval; as shown in Fig. 5 shows the time interval mostly falls into the certain time. This means, these 3-patterns are carried out in the fixed constant interval, and it can be considered as an evidence that these patterns of group A were sent from the same botnet system. With case of patterns $P_{1.3.49}$ and $P_{2.3.54}$, the averages of both patterns are greater than 14 *minutes* and these standard deviations are less than 6 *minutes*, but its histogram as shown in Fig. 5 shows time interval spreads and widely distributed. Therefore, we claim that 3-patterns of group B are the outcome of a collision of attacks have been made by some botnets.

IV. Attack Pattern Based on IP address and Timestamp

Botnet distributes malware toward the DSs through the Internet. Learn the behavior of the spreading malware from the source IP addresses and timestamps lead us to get the advantage as an alert from threats of the botnet's attack. For this purpose, we investigate both the source IP addresses have used by botnets and malwares' timestamps. First, we classify the sequential attack 3-patterns into several groups based on the source IP addresses and malwares' timestamps. Table V shows naming of the sequential attack 3-pattern based on source IP address, column *Source IP Pattern* is filled by the sequence of sources IP address of malware. Table VI shows naming of the sequential attack 3-pattern based on malwares' timestamps, column *Time Pattern* is filled by the time lines of honeypots were downloading the malware. For example, if $P_{1.3.27}$ has an Type of A_3E_3 as shown in Table IV, $i.e.$, the first and third malware are downloaded from the same source IP address (A_3), the second and third malware are downloaded at the same time (E_3).

Source IP addresses are discovered to distinguish the sources of malware. Some malware come from a single unique source IP address and some from many sources IP addresses. The unique host and pattern type of sequential attack 3-pattern can be seen in Table IV. Some sequential attack 3-patterns have single source IP pattern type, but some have two source IP pattern types. Groups of attacker A and B as mentioned before, have different source IP pattern types. The attacks have made by sequential attack 3-patterns in group A often use

TABLE V
Naming of the attack pattern based on source IP address

IP Pattern Code	IP Pattern		
A_1	$S1$	$S1$	$S1$
A_2	$S1$	$S1$	$S2$
A_3	$S1$	$S2$	$S1$
A_4	$S1$	$S2$	$S2$
A_5	$S1$	$S2$	$S3$

TABLE VI
Naming of the attack pattern based on timestamp

Time Pattern Code	Time Pattern		
E_1	$T1$	$T1$	$T1$
E_2	$T1$	$T1$	$T2$
E_3	$T1$	$T2$	$T2$
E_4	$T1$	$T2$	$T3$

source IP pattern types A_1, A_4, and E_1, whereas, the attacks have made by sequential attack 3-patterns in group B often follows close to the source IP pattern type A_3, A_5, and E_3. The sequential attack 3-patterns part of Group C are frequently downloaded from different source IP addresses (A_5) and the timestamp falls into the pattern E_3. Figure 6 and 7 shows the time charts of sequential attack 3-patterns $P_{1.3.29}$ and $P_{2.3.20}$ according to Tables V and VI to illustrate how botnets attacks based on source IP address and timestamp. Naming sequential pattern and mapping the behaviors of sequential attack patterns may give us knowledge about malware's spread, and lead us to identify and anticipate threats early.

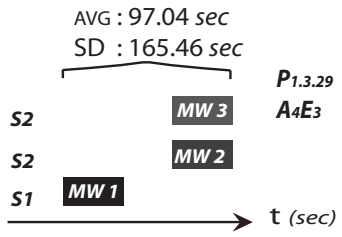


Fig. 6. Time chart of the sequential attack 3-pattern of malware has made by $P_{1.3.29}$

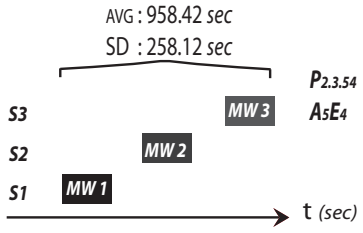


Fig. 7. Time chart of the sequential attack 3-pattern of malware has made by $P_{2.3.54}$

V. Conclusion

Our analysis shows that the coordinated attacks are performed by multiple sequential attack patterns within a short of time period. The sequential pattern of coordinated attacks tends to change all the time. The types of malware do not have a high dependency with operating system, but the sequence of malware within a pattern has operating system dependency.

This paper gives several behaviors useful for alerting threats of botnets' attacks. We have found that the *PrefixSpan* method sufficiently discover all sequential attack patterns.

Acknowledgment

We are greatful to AUN-SEED Net/JICA, which facilitated the Short-term Study Program in Japan.

References

[1] K. Kuwabara, et al., 'Heuristic for Detecting Botnet Coordinated Attack', in Proc. of the 4th International Workshop on Advances on Information Security (WAIS2010), 2010.

[2] M. Ohrui, H. Kikuchi, and M. Terada, "Mining association rules consisting of download servers from distributed honeypot observation", in IPSJ Malware Workshop (MWS2009), pp.151-156, 2009.

[3] O. Thonnard and M. Dacier, 'A framework for attack patterns' discovery in honeynet data', Digital Investigation, vol. 5, pp. S128-S139, 2008.

[4] W. Lina, et al., "Application of PrefixSpan* Algorithm in Malware Detection Expert System", in Education Technology and Computer Science (ETCS2009), pp. 448-452, 2009.

[5] R. Agrawal and R. Srikant, "Mining Sequential Patterns", in Data Engineering, 1995. Proc. of The 11th Int'l Conf. on Data Engineering (ICDE95), pp.3-14, 1995.

[6] R. Srikant and R. Agrawal,'Mining Sequential Patterns: Generalizations and Performance Improvements', in Proc. 5th Int. Conf. Extending Database Technology, (EDBT), pp.3-17, 1996.

[7] J. Pei, et al., "PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth", in Proc. of The 17th Int'l Conf. on Data Engineering, pp.215-224, 2001.

[8] F. Pedro "A survey on sequence pattern mining algorithms", in University of Informatics, Gualtar, Portugal, (http://alfa.di.uminho.pt/pedrogabriel/papers/SM_survey.pdf).

[9] Threat Encyclopedia. available from: ⟨http://threatinfo.trendmicro.com⟩.