Journal of Information Processing Vol.24 No.2 217–226 (Mar. 2016)

[DOI: 10.2197/ipsjjip.24.217]

Recommended Paper

TOPASE: Detection and Prevention of Brute Force Attacks with Disciplined IPs from IDS Logs

Satomi Saito[1,a) Koji Maruhashi[1] Masahiko Takenaka[1] Satoru Torii[1]

Received: May 8, 2015, Accepted: November [illegible], 2015

Abstract: Brute force attacks are used to obtain pairs of user names and passwords illegally by using all existing pairs to login to network services. These are a major security threat faced by network service administrators. In general, to prevent brute force attacks, administrators can set limitations on the number of login trials and shut down the traffic of brute force attacks with an intrusion prevention system (IPS) at the entry point to their services. In recent years, stealthy brute force attacks that can avoid the security rules and IPS and intrusion detection system (IDS) detection have appeared. Attackers tend to arrange a large amount of hosts and allocate them fewer login trials than the limitations administrators set. In this paper, we report a kind of distributed brute force attack event (brute force attacks with disciplined IPs, or *DBF*) against Remote Desktop Protocol (RDP) by analyzing IDS logs integrated from multiple sites. In *DBF*, a particular number of attacks is repeated automatically from a host to a service over a period. For this reason, existing countermeasures have no effect on *DBF*. We investigate the structure of *DBF* and improve the existing countermeasure system. We also propose TOPASE, which is replaced at each step of the existing countermeasure system and is suitable for *DBF* countermeasures. TOPASE analyzes the regularity of login trials between a source host and a destination host. Furthermore, TOPASE intercepts the network traffic from the source host of the brute force attack for a specific period. As a result of the evaluation with our IDS log, we estimate the performance of TOPASE and clarify the factors that maximize TOPASE's effectiveness.

Keywords: Remote Desktop Protocol, intrusion detection, log analysis, brute force attack

1. Introduction

The brute force attack is one of the many security threats that network service administrators must manage. Its method is to obtain pairs of user names and passwords illegally by trying all existing pairs to login to network services. In general, to prevent brute force attacks, administrators employ the following two countermeasures: set limitations on the number of login trials and put the host whose traffic is malicious on a blacklist. With those rules, administrators can stop login trials from a host until the limitation if the host continues login trials. By registering that host on the blacklist, they deny future login trials. In addition, administrators can shut down the traffic of brute force attacks by placing intrusion prevention system (IPS) at the entry point of their services. Some detection mechanisms are based on a simple anomaly detection focusing on login trials per minutes and periods that login trials are made [1]*1. Those are effective for only brute force attacks with a huge amount of login trials and a long time as a human can't do.

However, distributed and stealthy brute force attacks have emerged in recent years that can avoid IPS and intrusion detection system, IDS security rules and detection. In these brute force attacks, attackers arrange innumerable hosts and allocate them fewer login trials than the limitations administrators generally set. According to reports, in 2013, the well-known contents management system, WordPress, was the target of massive brute force attacks [2], [3]. It appears that the brute force attacks included more than a million login trials from about 9,000 different servers. In the same year, a source code management system, GitHub, also fell victim to massive brute force attacks [4]. These attacks occurred over long periods from about 40,000 IP addresses. In Ref. [5], a kind of distributed and stealthy brute force attack event called *brute force attacks with ephemeral IPs*, (*EBF*) is reported. According to the report, *EBF* has the following structures: specific services have detected brute force attacks with few login trials synchronously from a host at the same time. After an interval, almost the same services attacked from another host. This pattern of brute force attacks took place from hosts discretely and repeatedly. Those source hosts have attacked only once in the IDS log authors analyzed and they had a short life as *Ephemeral*. In Ref. [5], a countermeasure system against *EBF* is proposed. This system consists of two steps; extraction steps on IDS log analyses and shut down with prior monitoring.

Here, we report, to the best of our knowledge, the first analy-

1 FUJITSU LABORATORIES LTD., Kawasaki, Kanagawa 211–8588, Japan

a) sa.satomi@jp.fujitsu.com

The preliminary version of this paper was published at Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO 2014), July 2014. The paper was recommended to be submitted to Journal of Information Processing (JIP) by the chief examiner of SIGCSEC.

*1 In preventing brute force attacks with IPSs, the lower limit of login trials is higher than a human error and auto login facilities to control mass generation of false positives. In this paper, to extract the stealthy brute force attacks described below, we use IDS logs with set parameters so that fewer login trials are recorded as brute force attacks.

sis of another type of distributed and stealthy brute force attack against the Remote Desktop Protocol (RDP). This event has a different structure than that of *EBF* and cannot be detected and prevented by the *EBF* countermeasures described in Ref. [5]. By integrating real IDS logs detected from multiple sites, we gain an understanding of the advanced brute force attack event against RDP. This brute force attack event does not target plural services synchronously at the same time. Instead, a specific number of attacks are repeated automatically from a host to a service over a period. At a glance, those events appear to be human errors and auto logins. However, all the login trials between source and destination hosts share the same behavior although sources are different. Thus, we assert that this event is a kind of brute force attack caused by manipulated hosts. We name these attacks *brute force attacks with disciplined IPs* (*DBF*). We call these disciplined IPs due to the regularity of the login trials among each of the source hosts. Furthermore, we analyze login trial statistics, the co-occurrence of attack events in source/destination hosts and the relationships between source and destination hosts.

We also present a countermeasure system against *DBF*, TOPASE, which improves on the existing *EBF* countermeasure system. The existing *EBF* countermeasure consists of two steps: extracting victim hosts and shutting down suspicious traffic to the victims. However, *DBF*-targeted hosts cannot be extracted with just the extraction step because the *EBF* countermeasure algorithm is based on the synchronization between sources and destinations. Thus, we present TOPASE, a countermeasure system against *DBF*, improved each step suitable for detecting and mitigating *DBF*. TOPASE analyzes the login trial regularity between a source host and a destination host. The victim hosts share the regularity although the source hosts are different. The victim hosts are seen as being monitored more carefully than others because attackers focus on the *DBF* target for a while. Then, the shutdown step monitors the occurrence of the beginning of the next *DBF*. In the event of occurrence of the beginning of *DBF*, TOPASE intercepts the network traffic from *DBF* source host for a specific period. Therefore, TOPASE can detect the *DBF* victim hosts and prevent *DBF* from reaching the target hosts. We also evaluate the effectiveness of TOPASE with our IDS log. As a result, by intercepting traffic suddenly, TOPASE can intercept many brute force attacks that includes a *DBF* sequence even if the interception period is shorter than the periods that entire *DBF*s. In this case, the decreased usability is also minimized. Furthermore, TOPASE maintains a high performance once *DBF*s are analyzed and parameters are set.

Our contributions are as follows. First, by integrating our real IDS log from multiple sites, we report a distributed and stealthy brute force attack event with *DBF* occurring in multiple network services of RDP. Second, we present TOPASE, the system for detecting *DBF* victim hosts from IDS logs. TOPASE is constructed based on our analyses of real IDS logs. Third, we evaluate the effectiveness of TOPASE with our IDS log. As a result, we estimate the optimal TOPASE parameters and show that TOPASE maintains a high performance once *DBF*s are analyzed and parameters are set.

In Section 2, we describe reports related to brute force attacks and related works. In Section 3, we investigate the brute force attack event, *DBF* in detail. Section 4 addresses problems in applying *EBF* countermeasures to *DBF*s and presents TOPASE for detecting *DBF*. In Section 5, we evaluate the effectiveness of TOPASE with our real IDS log. Section 6 discusses setting optimal thresholds in TOPASE, applying TOPASE to *EBF* and the limitation of TOPASE. Finally, we conclude in Section 7.

Table 1 Example of IDS records.

srcIP	*dstIP*	*date*	*signature*	*count*
x.x.x.x	a.a.a.a	1/1 13:00	brute force (80/tcp)	10
y.y.y.y	b.b.b.b	1/2 14:00	brute force (80/tcp)	50
z.z.z.z	c.c.c.c	1/3 14:30	host sweep (10/tcp)	1
:	:	:	:	:

Here, *srcIP* means source IP addresses and *dstIP* means destination IP addresses. According to the algorithms of the IDS we use, *srcIP*s are detected as specific attacks, for example brute force attacks against a *dstIP*. In brute force attacks, IDS counts the number of login trials as *count*. The IDS log consists of the set of IDS records (record). A record shows that a *dstIP* detected an attack by a *srcIP* on a specific date. We show an example of IDS records in **Table 1**.

2. Related Works

2.1 Detecting Brute Force Attacks

There are many studies on detecting brute force attacks. Najafabadi et al. investigated several kinds of machine learnings for detecting brute force attacks with real data sets in Ref. [6]. Furthermore, not limited to brute force attacks, intrusion detection with anomaly detection and machine learning has been described in Refs. [7] and [8]. Hoque et al. tuned IDS parameters with a genetic algorithm and evaluated it with the KDD cup 1999 dataset in Ref. [8].

Tactical brute force attacks can be detected with network sequence level analyses. In Ref. [9], Hellemons et al. focused on brute force attacks against the secure shell (SSH) protocol consisting of three phases. They proposed a detection method and presented a prototype system. Satoh et al. proposed a technique for detecting dictionary attacks against SSH based on the relationship between a packet type and the data size in Ref. [10]. Mobin et al. presented a strategy for detecting stealth activities on networks and applied the dataset included in the distributed brute force attack event in Ref. [11]. Regarding intrusion detection, many studies adopt the network flow level approach, such as flow-based intrusion detection described in Ref. [12].

There also exists works related to detecting malicious scanning and interactions with regularity of network traffic in order to detect worm and botnet activities. In Ref. [13], Gu et al. proposed a system for detecting botnet C & C channels. Their method detects malicious interactions on IRC with the consistency of message sending. In Ref. [14], Malan et al. focused on the consistency of system calls invocations and proposed a method for detecting worms using P2P networks. Zhao et al. have proposed an approach for botnet traffic activity detection in Ref. [15]. In this study, they have also conducted a feasibility study of detecting botnet activities by classifying behavior based on time intervals.

2.2 Brute Force Attacks against RDP

Brute force attacks against RDP have been increasing in recent years. According to a report [16] and a study [17], the Morto worm has been infecting Windows workstations and servers through RDP since 2011. Morto launches login attempts for Remote Desktop servers with administrator privileges and a series of passwords[*2]. Furthermore, brute force attacks against RDP have been increasing according to an anti-virus software vendor's report [18] and topics about brute force attacks are addressed in a recent security monitoring report [19]. One report [20] indicates the sale of hacked RDP installations and shows the account list that is frequent in RDP, but attackers can potentially get access to all system information.

On the Internet, there exists many available tools for brute force attacks against RDP. Users of those tools set user names and password list files and start brute force attacks to target hosts. They can also delay connection times, login attempts and so on. Most tools provide a GUI to start brute force attacks easily. NCRACK [21] and THC-Hydra [22] are open source brute force attack tools. They support well-known protocols like SSH, FTP, HTTP and RDP. Many blogs and news sites show the benchmarks and comment on these tools. Furthermore, Brutik RDP [23] and TSGrinder [24] are also available for the tools specialized for RDP brute force attacks. User names and password lists, of course, can be obtained easily. Not only are those lists open to the public, but there are also lists tagged effectively for brute force attacks against RDP. Therefore, many resources for brute force attacks against RDP are accessible today.

2.3 Brute Force Attack with Ephemeral IPs (*EBF*) and its Countermeasures

We describe the structure of *EBF* and the countermeasure system against *EBF* in Ref. [5]. *EBF* is a kind of distributed brute force attack as **Fig. 1**. Several specific *dstIP*s have been the target of *EBF* and have been detected as brute force attacks on the same date and login trial. For each brute force attack, about a dozen login trials per *dstIP* are attempted for several minutes. Such sequences of login trials have been detected repeatedly, with intervals. On the other hand, the *srcIP*s are a large amount and not reused. Authors also show this after analysis of an *EBF* event. Adversaries who intend to launch brute force attacks have used a large amount of IP addresses as a method of camouflaging their attacks. For each victim of *EBF*, they cannot determine the occurrence of this attack. They see only that about a dozen login trials occurred through many unique *srcIP*s. In many cases, it is common for most network services to be accessed from many unique hosts even if some login trials failed.

Figure 2 shows the countermeasure system presented in Ref. [5]. This countermeasure consists of two steps: extracting *dstIP*s that are victims of *EBF*s from IDS logs and shutting down *EBF*s to the victims by monitoring traffic to them and detecting *EBF*s. At first, at the extraction step, the *dstIP*s victim of *EBF*s are extracted from IDS logs accumulated over a specific

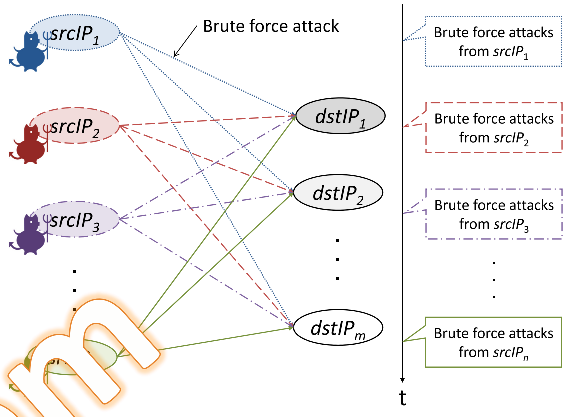

Fig. 1 Example of *EBF* architecture.

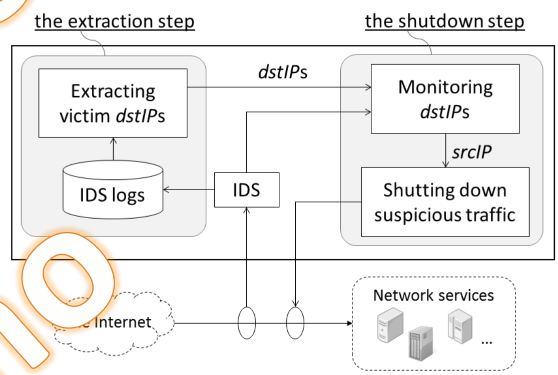

Fig. 2 *EBF* countermeasure system architecture.

period. The IDS logs are analyzed based on the correlation between *dstIP*, *srcIP* and the detected date. The shutdown step uses the synchronization between *srcIP* and plural *dstIP*s that *EBF* has. If brute force attacks are detected from a *srcIP* to *dstIP*s included as victims, the system suspects the occurrence of part of *EBF* and shuts down the traffic from the *srcIP* for a specific period. For example, $dstIP_1$, $dstIP_2$ and $dstIP_3$ are extracted at the extraction step. An unknown $srcIP_1$ try to launch a brute force attack to $dstIP_1$, $dstIP_2$ and $dstIP_3$. The countermeasure system monitors and detects $srcIP_1$ as a member of *EBF*. Subsequently, the system shuts down the traffic from $srcIP_1$ for several minutes.

3. Brute Force Attacks with Disciplined IPs (*DBF*)

In this section, we identify another type of brute force attack event described above as *brute force attacks with disciplined IP addresses* (*DBF*). We investigate its features with respect to improving the existing countermeasure system and provide three viewpoints; login trial, *srcIP* and *dstIP*.

3.1 *DBF* Structure

DBF is a kind of distributed brute force attacks as **Fig. 3** shows. In *DBF*, plural *srcIP*s attempt to log in to targeted *dstIP*s for specific periods and the frequency of login trials is constant. Our IDS logs indicates that each *srcIP* sets a *dstIP* as a target of brute force attacks. In this paper, the *DBF* sequence indicates that a *srcIP* continues login trials to a *dstIP* for a specific period. A *DBF* sequence consists of plural brute force attacks between a *srcIP* and a *dstIP* (a *srcIP* and *dstIP* pair). A brute force attack

[*2] We guess that our IDS logs include the behaviors by Morto. However, for the identification of Morto, it is required for collaborating IDS log and another type of log, for example, TCP dumps.

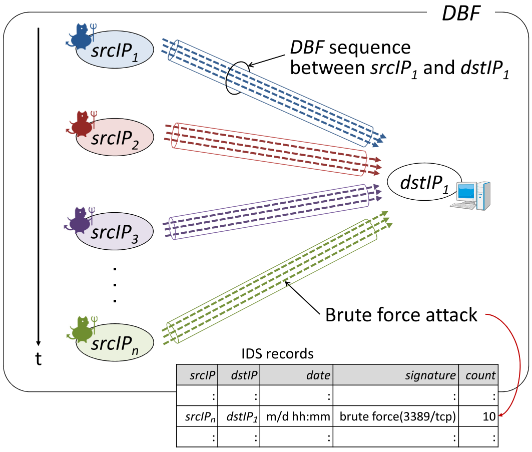

Fig. 3 Example of *DBF* architecture.

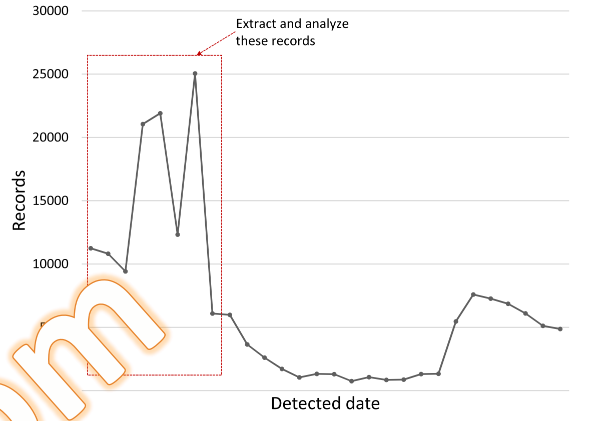

Fig. 4 The amount of records detected as brute force attacks against RDP in 28 months.

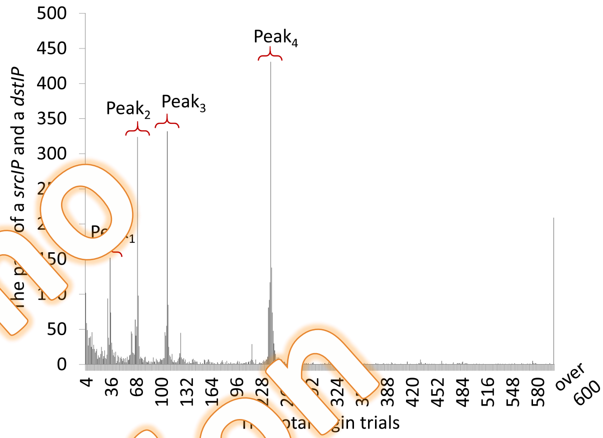

Fig. 5 Distribution of total number of login trials for every *srcIP* and *dstIP* pair.

is detected by IDS and recorded as one record on the IDS log. In Fig. 3, $srcIP_1, srcIP_2, srcIP_3, \ldots, srcIP_n$ are the *srcIP*s under *DBF*, and their target is $dstIP_1$. This figure includes n *DBF* sequences, a *DBF* sequence between $srcIP_1$ and $dstIP_1$, $srcIP_2$ and $dstIP_1$, $srcIP_3$ and $dstIP_1$, ..., $srcIP_n$ and $dstIP_1$. Those *DBF* sequences are executed toward the targeted $dstIP_1$ along a time series.

We describe the difference between *EBF* and *DBF*. In *EBF*, a *srcIP* of *EBF* executes brute force attacks to plural *dstIP*s at the same time. Those targeted *dstIP*s have also a correlation in the number of login trials. On the other hand, the definition of *DBF* is focused on the regularity of brute force attacks between *srcIP* and *dstIP*. The *srcIP*s of *DBF*s continue brute force attacks to a *dstIP* with the same number of login trials among each brute force attack.

We have the conviction that the *DBF* is caused by attackers who have many IP address resources and manipulates them. However, we state the possibility that some of *DBF* sequences contain the following cases. First, legitimate users send wrong passwords. If they keep on sending wrong passwords until IDSs detect as brute force attacks, that behavior is similar to that of *DBF* sequences. Second, multiple kinds of malware happen to try to login to the same target. In this case, it looks like those malwares cooperated with each other as included *DBF*. Third, malicious human attackers also try to login to the target at almost the same time.

3.2 Analyzing inside of *DBF*

To gain the features beneficial for the *DBF* countermeasure, we investigate the statistical behavior around *DBF* with our IDS log. We observe the traffic suspected of being a brute force attack against RDP and show the number of recorded brute force attacks against RDP using IDS in **Fig. 4**. The horizontal axis is the detection dates for 28 months from 2011 to 2014, and the vertical axis is the number of records. From this figure, we extract 117,924 records detected in the first eight months to analyze *DBF* in detail. These extracted records contain 3,260 unique *srcIP*s and 53 unique *dstIP*s. In this subsection, we investigate the co-occurrence of detected signatures with respect to *srcIP* and *dstIP*. In this investigation, we eliminates two months obtaining six-months records from extracted eight-months records.

3.2.1 Login Trials

We set up a hypothesis about *DBF* as follows: *DBF srcIP*s show the same behaviors in login trials. That behavior are apparent through the statistics of the number of login trials. At first, we show the results of counting the number of login trials for every *srcIP* and *dstIP* in the distribution in **Fig. 5**. The horizontal axis shows the total number of login trials, and the vertical axis shows the number of *srcIP* and *dstIP* pairs. Disregarding the minimum and those over 600 in the total number login trials, this figure has four peaks labeled $Peak_1$, $Peak_2$, $Peak_3$ and $Peak_4$. There exists much *srcIP*s whose total number of login trials are equal to those peaks. Those peaks show that the brute force attacks whose *srcIP*s share a login trial behavior. Next, for the *srcIP*s of those peaks, we investigate the regularity of brute force attacks between a *srcIP* and *dstIP* pair in a *DBF* sequence. **Figure 6** shows the relationships between the average number of trials to login per record and the standard deviations of the number of trials for *srcIP* and *dstIP* pairs. The standard deviation (*STD*) of the number of trials appears on the vertical axis. Many circles are within the same range on each chart. The range of the average number of trials is from about 5 to 10, and standard deviations of the number of trials are from about 0 to 2. Each *DBF* sequence shares the same average number of trials to login and standard deviations of the number of trials. Furthermore, to confirm the regularity of each peak in detail, we focus on the duration of a brute force attack from a *srcIP* to a *dstIP*. We compare the durations of brute force attacks for every *srcIP* and *dstIP* pair in **Fig. 7**. Considering the IDS algorithms, we count the duration of brute force attacks if the difference in the time detected in the records is less than 3 minutes. In Fig. 7, the distances for each peak in Fig. 7 are similar to those in Fig. 5.

Therefore, in our IDS log, there exists *DBF* events whose *srcIP*s and *dstIP*s share the same login trial behavior. Each *srcIP* continues login trials with the same number. The difference in login trials comes from the duration of a *DBF* sequence. Furthermore, the login trial regularity is the beneficial feature for extracting and mitigating *DBF* with countermeasures. Even if each *srcIP* is unique, login trials continue at the same rate in a specific period.

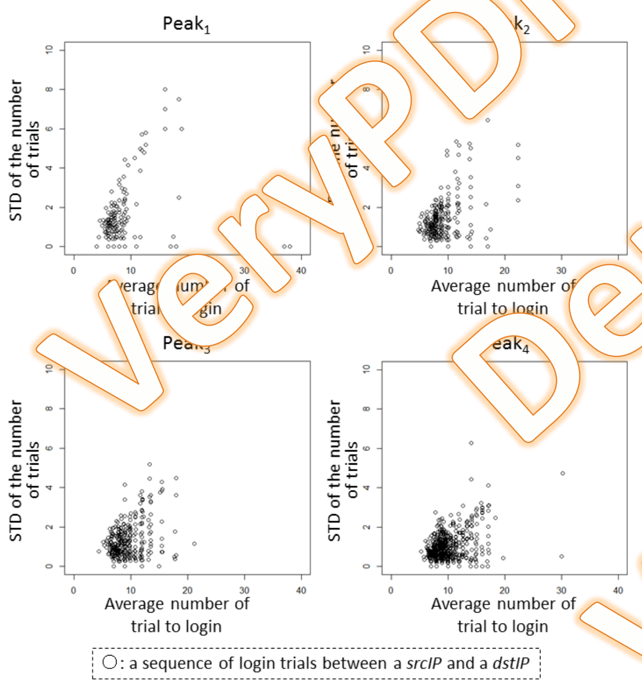

Fig. 6 Relationship between the average number of trials to login and the standard deviation of the number of trials on each Peak.

3.2.2 *srcIP* and *dstIP*

We investigate with respect to *srcIP* and *dstIP* and measure the frequency of *srcIP* detection and co-occurrence in IDS records. For the investigation focusing on *DBF*, we extracted 11,982 records related to *DBF* sequences based on the investigation of login trials.

We plot the extracted records of *srcIP*s, *dstIP*s and detected dates in **Fig. 8**. In this figure, the horizontal axis is the detected dates in an eight-month periods and the vertical axis is the unique *dstIP*s. The dot shape depends on the unique *srcIP*. Some *dstIP*s are detected from single *srcIP*s repeatedly and others from a large amount of *srcIP*s. In *DBF*, *dstIP*s were detected in brute force attacks from only several *srcIP*s. Thus, to investigate *srcIP* reuse frequency, we count unique days when *srcIP*s were recorded by IDS. As a result, over 93% of *srcIP*s in *DBF* are recorded for only one day in our IDS log. Almost all others are recorded for nearly two days. However, there exists two *srcIP*s recorded in two sequences; those account for about 0.6% of *srcIP*s in all *DBF* sequences. Therefore, almost no *srcIP* are not reused on a given

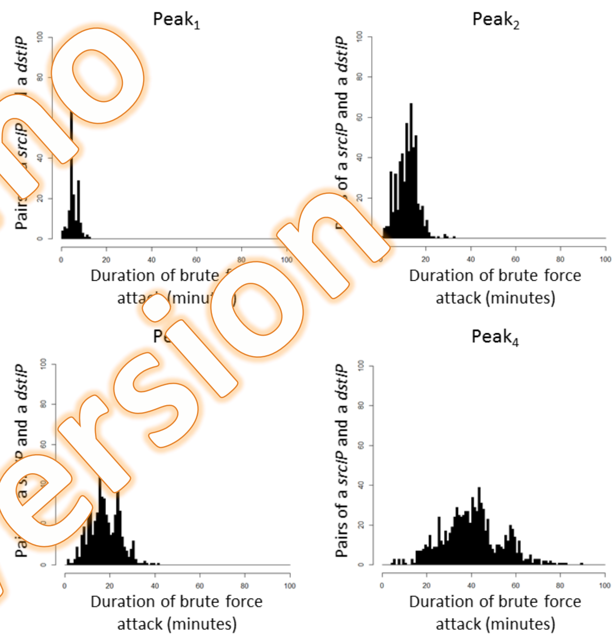

Fig. 7 Duration of brute force attack for every *srcIP* and *dstIP* pair.

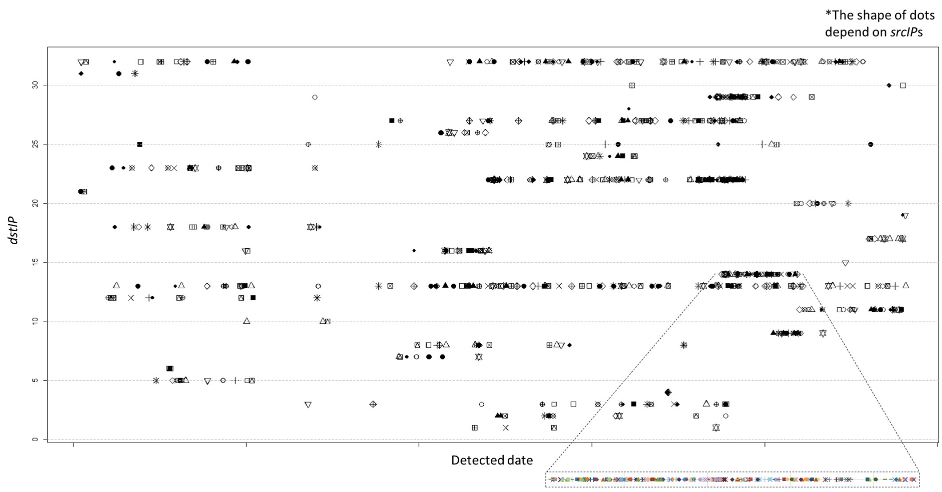

Fig. 8 Relationship between detected dates and *DBF*s.

Table 2 Detected Reasons co-located with RDP brute force attacks in *srcIPs* (Top 5).

	Detected Reason	*srcIPs* (%)
all	Host sweep (3389/tcp)	1.81
	Netbios scan (137/udp)	0.77
	Host sweep (1433/tcp)	0.31
	SMB (Netbios) service scan (445/tcp)	0.18
	SMB service connect (445/tcp)	0.12
DBF	Netbios scan (137/tcp)	1.76
	Host sweep (3389/tcp)	0.35
	Malicious scanning (40235/tcp)	0.35
	DNS ETC type query flooding (53/udp)	0.18
	Host Sweep (1433/tcp)	0.18

Table 3 Detected Reasons co-located with RDP brute force attacks in *dstIPs* (Top 5).

Detected Reason	*dstIPs* (%)
Netbios Scan (137/udp)	100
MSSQL Server2000 Resolution Service DOS (1434/udp)	100
Slamer Worm (1434/udp)	100
SMB Service sweep (445/tcp)	79.25
Trace Route (0/tcp)	77.36

day.

Next, we investigate the co-occurrence in the IDS record. In some cases, scanning activities are detected as a sign of the malicious launch of brute force attacks. **Table 2** shows the list of reasons detected and their percentages of *srcIPs* detected as brute force attacks and *DBF* against RDP. In respect from the brute force attack, it is clear that there are few *srcIPs* detected simultaneously in some attacks involving brute force attacks against RDP. On *DBF*, only about 1.76% of *srcIPs* are detected as brute force attacks and Netbios scan. From this result, we cannot determine the correlation between brute force attacks and any other attacks on the *srcIP*. On the *dstIP*, **Table 3** shows the list of reasons detected and their percentages of *dstIPs* detected as brute force attacks against RDP. In contrast to *srcIP*, most *dstIPs* are detected in not only brute force attacks, but also in other attacks related to Windows services. We also investigate the co-occurrence of an IDS record with respect to *dstIP* of *DBF*. However, the result is the same as that of Table 3. Therefore, there also exists no difference between *DBF* and entire brute force attacks. Our co-occurrence investigation cannot identify any co-occurrence that distinguishes *DBF* from other brute force attack events.

Thus, it is difficult to derive any features beneficial to *DBF* countermeasures from the viewpoint of *srcIP* and *dstIP*.

4. Countermeasure against *DBF*: TOPASE

In this section, we present TOPASE, the countermeasure system against *DBF* described above. We improve the existing countermeasure system shown in Fig. 2 to be suitable for extracting and mitigating *DBF*. According to the investigation of records related to *DBF* events, *srcIPs* and *dstIPs* pair share the same login trials behavior and *srcIPs* continue login trials at the same rate for a specific period. Therefore, TOPASE extracts *DBF* victim *dstIPs* based on the regularity of login trials and shuts down suspicious traffic to *DBF* by detecting the launch of brute force attacks, whose trials are subject to rules.

4.1 Problem in Applying the Existing Countermeasure to *DBF*

In applying the *EBF* countermeasure system to *DBF*, there are several problems at the extraction and shutdown steps. Those problems are caused by asynchronism between the *dstIP*, *srcIP* and the detected date.

At the *EBF* countermeasure extraction step, a cross tabulation of login trials is first estimated among *dstIPs*, *srcIPs* and detected dates. From the cross tabulation table, *dstIPs* with high correlations are extracted, as they have detected brute force attacks from the same *srcIP* on the same date. Those *dstIPs* are the *EBF* targets. On *DBF*, however, the victim *dstIPs* are not detected as brute force attacks from the same *srcIP*. For this behavior, no *dstIPs* are extracted even if calculating correlations between *dstIPs*. At the shutdown step, the countermeasure system monitors the traffic to the extracted *dstIPs* at the previous step. As a sequence of *EBF* caused by a *srcIP* reaches several targeted *dstIPs*, if some monitored *dstIPs* are detected as brute force attacks from a *srcIP*, the system shuts down the traffic from the *srcIP* for a specific period. In *DBF*, a *DBF* sequence of brute force attacks per *srcIP* reaches one of the *dstIPs* targets. Therefore, even if monitoring targeted *dstIPs* focus on the attack from a single *srcIP* to plural *dstIPs*, the countermeasure system misses the beginning of the sequence. From the above, using only existing *EBF* countermeasures, victim *dstIPs* cannot be extracted at the extraction step and beginnings of sequences cannot be detected at the shutdown step. Thus, another mechanism is required for *DBF*.

In the next section, we investigate the *DBF* features obtained from IDS logs described in this section and improve the current *EBF* countermeasure system.

4.2 Entire Architecture of TOPASE

The entire architecture of TOPASE follows the *EBF* countermeasure system shown in Fig. 2. TOPASE monitors the network traffic to an internal network with IDS. One or more sites operate network services in this internal network. IDS detects brute force attacks from inbound traffic and sends alerts to the extraction step in TOPASE as IDS records.

At first, the extraction step collects the IDS records and detects the *DBF* victim *dstIPs* based on the regularity of login trials. The extracted *dstIPs* are sent to the next step, the shutdown step. Next, at the shutdown step, these *dstIPs* are regarded as being monitored more carefully than others because attackers remain on the *DBF* target for a while. The shutdown step also receives the IDS records detected for the *dstIPs* and checks whether the brute force attack is the beginning of the next *DBF* sequence. If the beginning of next *DBF* sequence occurs, the shutdown step intercepts the network traffic from the *srcIP* of the brute force attack for a specific period.

By intercepting the network traffic suspected of being a *DBF*, TOPASE prevents the *DBF* sequence from reaching the target *dstIPs*, except for the beginning of the sequence. As the interception time is limited, the number of *srcIPs* to intercept does not increase infinitely like simple black lists.

4.3 Extraction Step: Extracting the Victim *dstIPs*

At the extraction step, the IDS records collected are analyzed to detect the *DBF* victim *dstIP*s. In *DBF*, each *DBF* sequence shares the same login trial behavior. The start of this step calculates several statistical parameters necessary to compare and record the same login trials behavior for all existing *srcIP* and *dstIP* pairs. The statistics include the number of total login trials, the average number of trials to login, the mode, the standard deviation of the number of trials and other factors. With the login trial properties of each pair, we describe several processes to detect such *DBF* sequences.

First, count the properties of *srcIP* and *dstIP* pairs in the total number of login trials and extract one and more peaks. The pairs included in *DBF* sequences share the same login trial behavior. If IDS records includes *DBF* sequences, specific values of properties are counted more times than others. Those values correspond to those of *DBF* sequences. We showed the effectiveness of this process above. We detected and extracted *DBF* sequences from our IDS log. Four peaks appear in the distribution in the number of total login trials in Fig. 5. Furthermore, when investigating distributions of the average number of trials to login and the standard deviation of the number of trials, it is expected that a peak appears around 10 in the average and 1 in the standard deviation of the number of trials. In this process, system users set the threshold at which peaks are extracted from distributions.

Second, apply clustering tools to all pairs and create subsets based on the similarity of their properties. The clustering tools input vectors as the values of the properties. The clustering tools available include, for example, k-means [25] and self-organizing maps (SOMs) [26]. If *DBF* sequences occur and their login trials are similar, the pairs corresponding to *DBF* are clustered into specific subsets because those pairs share the same or similar properties. Furthermore, those subsets contain higher numbers of pairs than other subsets if *DBF* sequences last for specific terms. Under those conditions, *DBF* sequences are extracted by selecting up the subset with a larger number of pairs than the others. In this process, *DBF* sequences are extracted more precisely even if their properties are not accurately equal to the peaks in the distributions. In addition, the thresholds are also needed in the number of subsets for clustering tools and for determining which large subset to select for extraction of *DBF* sequences.

4.4 Shutdown Step: Intercepting Network Traffic Suspected of *DBF*

At the shutdown step, the *dstIP*s extracted at the previous step are regarded as being monitored more carefully than others because attackers focus on the *DBF* target for a while. Subsequently, the traffic to the *dstIP*s is monitored using the IDS records detected as brute force attacks and the occurrence of the next *DBF* sequence is waited for.

In monitoring the IDS records detected as brute force attacks, the following record is the trigger of the interception of suspicious *DBF* sequences and traffic. The record is detected as a brute force attack, and the destination is included with the *dstIP*s extracted at the previous step. The number of login trials is similar to the value that *DBF* sequences have at the extraction step. If the records confirmed under those conditions are sent to this step, assume that the first *DBF* sequence has occurred and wait for the records in which *srcIP* and *dstIP* are the same and the number of login trials is also the same or similar to the trigger record. If the awaited record is sent to the step in a predefined interval several times, intercept the traffic from the *srcIP* for a specific period. The traffic from *srcIP* is the beginning of a *DBF* sequence intercepted and prevented from reaching the target *dstIP* until the *DBF* sequence finishes. After interception for a specific period, stop intercepting traffic and delete the trigger information.

At this step, system users determine the following four thresholds; 1) number of login trials and its range to trigger the suspicion of the first *DBF* sequence, 2) standard deviations of the number of trials suspected to be *DBF* sequence beginnings, 3) number of records until the start of traffic interception and 4) traffic interception period. From those thresholds, 1), 2) and 4) are determined from the results of the extraction step. The observed *DBF* sequence is analyzed at the previous step.

5. Evaluation of TOPASE

In this section, we evaluate the effectiveness of TOPASE. We simulate the shutdown step in TOPASE with our IDS log described in Section 4. We evaluate the effectiveness from the following two viewpoints. The first is the dropping rate, which indicates the rate at which TOPASE drops the traffic included in a *DBF* sequence. Small values of dropping rate show that TOPASE could intercept a large amount of brute force attacks of a *DBF* sequence. The second is the wasted period, which indicates how long TOPASE monitors and waits for intercepting the traffic although a *DBF* sequence has terminated. Small values of wasted period show that TOPASE could stop monitoring and intercepting the traffic as soon as a *DBF* sequence had terminated[*3].

We extract the following four record subsets detected in a month: 5th, 8th, 17th and 23rd in Fig. 4. The previous two months overlap the records extracted for detailed analysis. From the latter two months, we extract records corresponding to the peaks described in Fig. 6. We also show the relationship between the average number of trials to login and the standard deviation of the number of trials in **Fig. 9**. From the figure, *DBF*s were also detected in the two months. The scale of each month is shown in **Table 4**. For those four subsets, we measure the dropping rate and the wasted period of *DBF* sequences while changing the number of records until the interception start (S) and the interception period (P). In our evaluation, we set the two thresholds as follows according to the analyses in Section 3: the login trials is from 4 to 10 and the standard deviation of the number of login trials is from 0 to 2.

As a result of applying the shutdown step in TOPASE for four months, in each of the month, we compare the results changing $S = 1, 2, 3$ and $P = 20, 40, 60$ minutes. We show the averages of dropping rates and the wasted periods in **Figs. 10**, **11**, **12** and **13**. From the viewpoint of the dropping rate, about 12.3% of a

[*3] In this evaluation, the wasted period is not equal to a false positive. To estimate the false positive, it is required to perform TOPASE on the environment where the administrators can distinguish legitimate login trials from malicious trials.

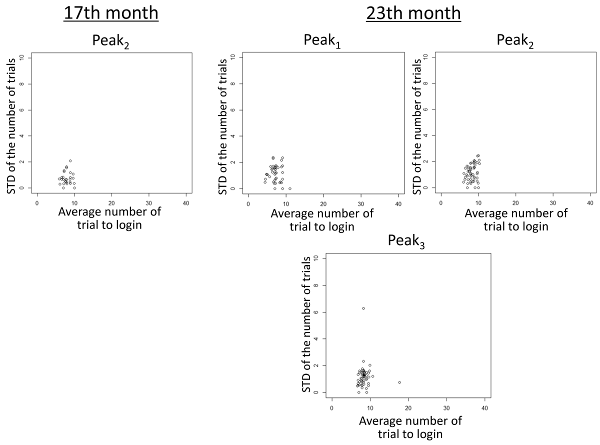

Fig. 9 Relationship between the average number of trials to login and the standard deviation of the number of trials on peaks extracted from the 17th and 23rd months.

Table 4 The scale of four subsets of IDS records.

Month	Records	unique *srcIP*	unique *dstIP*
5th	21,905	709	21
8th	6,095	366	21
17th	1,068	[illegible]	21
23rd	7,591	52	19

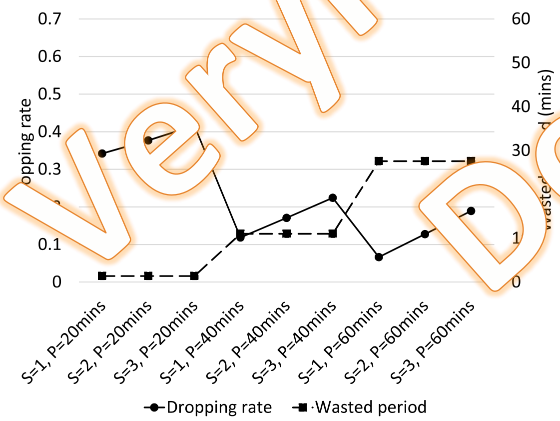

Fig. 10 Dropping rate and wasted period in the 5th month.

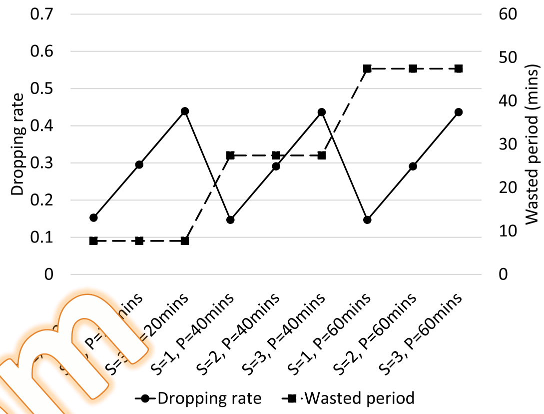

Fig. 11 Dropping rate and wasted period in the 8th month.

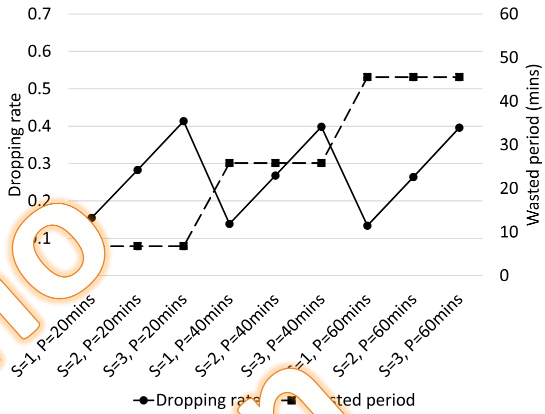

Fig. 12 Dropping rate and wasted period in the 17th month.

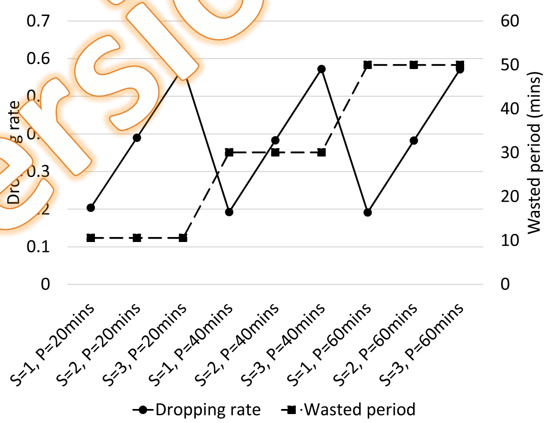

Fig. 13 Dropping rate and wasted period in the 23rd month.

DBF sequence on average is passed if TOPASE considers and waits to start the interception of one more alerts. The dropping rate increase is at its lowest in the 1st month. This indicates that the 1st month includes the *DBF* sequences that maintain brute force attacks longer than in the other three months. Detecting the beginning of a *DBF* sequence is more critical in the other three months than in the 1st month. From the viewpoint of the wasted period, tuning P to 60 minutes wasted more periods than others although this case covers *DBF* sequences even when their brute force attack periods are long. Not counting the results in the 1st month, the wasted period increase is about 19.4 minutes on average when P changes from 20 to 40. The wasted period increase is about 19.9 minutes on average when P changes from 40 to 60. These results show that increments of interception periods become a direct cause of the wasted period.

From these evaluations, we learn pieces of information. First, to the best of our knowledge, the optimal parameters are $S = 1$ and $P = 20$. Tuning P to be excessively long increases the wasted periods and decreases the usability of network services to users. When a user fails to log in to a server a specific number of times, IDS detects the behavior as a brute force attack. If TOPASE receives these alerts and starts intercepting the traffic from the user, the user cannot access the server until the interception is over. If a *DBF* sequence has a longer period than the tuning, TOPASE restarts interception of this sequence after 20 minutes of interception, although several brute force attacks occurred between the finish and restart of the interception. Furthermore, tuning P to a short period has another merit. TOPASE monitors and examines every IDS alert related to brute force attacks although a *DBF* sequence has terminated. This causes TOPASE performance degradation. Second, we learn that TOPASE does not require frequent

S and P tuning. The results of four months consistency show that the optimal S and P parameters are $S = 1$ and $P = 20$. Therefore, it is possible to continue operating TOPASE under the best conditions once administrators set those parameters. In addition, we discuss the consistency of other parameters in the next section.

6. Discussion

6.1 Applying TOPASE to *EBF*

As described in Section 4, there are several problems in applying the *EBF* countermeasure system to *DBF*. We discuss applying TOPASE to *EBF*.

In *EBF*, a *srcIP* of *EBF* executes brute force attacks against plural *dstIP*s at the same date with the same number of login trials among them. The *EBF* countermeasures can detect the targeted *dstIP*s based on the high correlations in the detected date and the number of login trials. On the other hand, TOPASE detects targeted *dstIP*s which continue brute force attacks with a login trial regularity. Hence, TOPASE cannot detect the *EBF*-targeted *dstIP*s because the analysis of high correlations among *dstIP*s is out of range for TOPASE.

The shutdown step in TOPASE also focuses on the login trial regularity. TOPASE pays no attention to whether attack targets are registered or to synchronizations among targets. TOPASE passes the *EBF* traffic unless the *EBF* continues a brute force attack with regularity.

6.2 Setting Optimal Thresholds in TOPASE

To monitor and intercept a traffic corresponding to *DBF*, it is important to set optimal thresholds at the TOPASE extraction step. At this step, it is also necessary for human administrators to tune several parameters. In the simplest case, administrators set TOPASE thresholds referring directly to plots and various statistics, such as the evaluation described above. Besides, when using clustering tools, users must consider parameters such as which cluster is needed.

However, in our 28-month monitoring, *DBF* sequences were detected with their statistics remaining the same. The number of *DBF* sequences increases or decreases in a time series, and the averages of login trials and the standard deviations of the number of trials are shown in Fig. 9. Therefore, once human administrators refer to the statistics of traffic suspected to be *DBF* and extract the parameters for the shutdown step, TOPASE is able to continue intercepting a *DBF* traffic without maintenance or applying the extraction step continuously.

6.3 Attackers Aware of TOPASE

We discuss the limitations of TOPASE against attackers who are aware of TOPASE countermeasures To avoid TOPASE, attackers randomize frequency and the total number of login trials of each *DBF* sequence. Current TOPASE focuses on the regularity of login trials common to each *DBF* sequence. To avoid the detection of this regularity, the attacker allocates a different number of login trials and interval times at random. As a result, TOPASE passes the randomized *DBF* sequences. To mitigate this weakness, the inspection of the randomness of a *DBF* sequence is required. If some brute force attacks are detected by IDSs, TOPASE distinguishes a *DBF* sequence with randomized interval and login trials from all brute force attacks and starts the interception. In distinguishing random *DBF* sequences, it takes more alerts than standard *DBF* sequences until determining and starting *DBF* sequences interception. As related works, in Ref. [27], Wu et al. proposed a method for detecting scanning by malwares. Their method also detects random scanning with a distribution of their properties.

7. Conclusion

In this paper, we report a type of distributed and stealthy brute force attack event, called brute force attacks with disciplined IPs (*DBF*) from our real IDS log. By integrating real IDS logs detected from multiple sites, we find that each login trial between a *srcIP* and *dstIP* share the same behavior, although sources are different. This is a clue to detect distributed and stealthy brute force attack events. We also present a countermeasure against *DBF*, called TOPASE, which improves the existing countermeasure system. TOPASE analyzes the regularity of login trials between a source host and a destination host at the extraction step. The shutdown step monitors the occurrence of the beginning of next *DBF* sequences. If the beginning of a *DBF* sequence takes place, TOPASE intercepts the network traffic from the source host of the brute force attack for a specific period. As a result of the evaluation of the shutdown step in TOPASE with our IDS log, by starting the traffic interception suddenly, TOPASE can intercept many brute force attacks, including *DBF* sequences, even if the interception period is shorter than the periods that can cover entire *DBF*s. In this case, the usability decrease is also minimized. Furthermore, TOPASE maintains a high performance once *DBF*s are analyzed and parameters are set.

References

[1] Lazarevic, A., Banerjee, A., Chandola, V., Kumar, V. and Srivastava, J.: Data Mining for Anomaly Detection, *Tutorial at the European Conference on Principles and Practice of Knowledge Discovery in Databases* (2008).

[2] SUCRI Blog: Mass WordPress Brute Force Attacks - Myth or Reality (2013), available from ⟨http://blog.sucuri.net/2013/04/mass-wordpress-brute-force-attacks-myth-or-reality.html⟩.

[3] SUCRI Blog: The WordPress Brute Force Attack Timeline (2013), available from ⟨http://blog.sucuri.net/2013/04/the-wordpress-brute-force-attack-timeline.html⟩.

[4] PCWorld: GitHub bans weak passwords after brute-force attack results in compromised accounts (2013), available from ⟨http://www.pcworld.com/article/2065340/github-bans-weak-passwords-after-bruteforce-attack-results-in-compromised-accounts.html⟩.

[5] Honda, S., Unno, Y., Maruhashi, K., Takenaka, M. and Torii, S.: Detection of Novel-Type Brute Force Attacks used Ephemeral Springboard IPs as Camouflage, *International Conference on Information and Network Security* (*ICINS2014*) (2014).

[6] Najafabadi, M.M., Khoshgoftaar, T.M., Kemp, C., Seliya, N. and Zuech, R.: Machine Learning for Detecting Brute Force Attacks at the Network Level, *IEEE International Conference on Bioinformatics and Bioengineering* (*BIBE*), pp.379–385 (2014).

[7] Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G. and Vazquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges, *ELSEVIER Computers & Security*, Vol.28, No.1-2, pp.18–28 (2009).

[8] Hoque, M.S., Mukit, M.A. and Bikas, M.A.N.: An Implementation of Intrusion Detection System Using Genetic Algorithm, *International Journal of Network Security & Its Applications*, Vol.4, No.2, pp.109–120 (2012).

[9] Hellemons, L., Hendriks, L., Hofstede, R., Sperotto, A., Sadre, R. and Pras, A.: SSHCure: A Flow-Based SSH Intrusion Detection System, *6th IFIP WG 6.6 International Conference on Autonomous Infrastruc-*

ture, Management, and Security (*AIMS 2012*), pp.86–97 (2012).
[10] Satoh, A., Nakamura, Y. and Ikenaga, T.: SSH Dictionary Attack Detection Based on Flow Analysis, *IEEE/IPSJ 12th International Symposium on Applications and the Internet* (*SAINT2012*), pp.51–59 (2012).
[11] Javed, M. and Paxson, V.: Detecting Stealthy, Distributed SSH Brute-forcing, *2013 ACM SIGSAC Conference on Computer & Communications Security*, pp.85–96 (2013).
[12] Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. and Stiller, B.: An Overview of IP Flow-Based Intrusion Detection, *IEEE Communications Surveys & Tutorials*, Vol.12, No.3, pp.343–356 (2010).
[13] Gu, G., Zhang, J. and Lee, W.: BotSniffer: Detecting botnet command and control channels in network traffic, *15th Annual Network and Distributed System Security Symposium* (*NDSS2008*) (2008).
[14] Hyunsang, C., Lee, H. and Kim, H.: BotGAD: Detecting botnets by capturing group activities in network traffic, *4th International ICST Conference on Communication System Software and Middleware*, ACM (2009).
[15] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A. and Garant, D.: Botnet detection based on traffic behavior analysis and flow intervals, *ELSEVIER Computers & Security*, Vol.39, Part A, pp.2–16 (2013).
[16] F-Secure: Windows Remote Desktop Worm "Morto" Spreading - F-Secure Weblog: News from the Lab, available from ⟨http://www.f-secure.com/weblog/archives/00002227.html⟩ (accessed 2015-04-09).
[17] Martin, V. and Vykopal, J.: Flow-based detection of RDP brute-force attacks, *7th International Conference on Security and Protection of Information* (*SPI 2013*) (2013).
[18] Security Affairs: Kaspersky Lab reveals an increase in RDP bruteforce attacks - Security Affairs, available from ⟨http://securityaffairs.co/wordpress/26247/cyber-crime/kaspersky-lab-reveals-increase-rdp-bruteforce-attacks.html⟩ (accessed 2015-07-09).
[19] Alert Logic: CLOUD SECURITY REPORT- SPRING 2014, pp.3–7 (2014).
[20] Hacked Via RDP: Really Dumb Passwords ? Krebs on Security, available from ⟨http://krebsonsecurity.com/2013/12/hacked-via-rdp-really-dumb-passwords/⟩ (accessed 2015-03-09).
[21] Ncrack - High-speed network authentication cracker, available from ⟨http://nmap.org/ncrack/⟩ (accessed 2015-03-09).
[22] THC-HYDRA - fast and flexible network login hacker, available from ⟨https://www.thc.org/thc-hydra/⟩ (accessed 2015-03-09).
[23] Brutik RDP - Special edition [CRACKED] - Don't Mess With Your Best Leave The Rest, available from ⟨http://pk6...t.blogspot.jp/2012/08/brutik-rdp-special-edition-cracked.html⟩ (accessed 2015-03-09).
[24] HoG Downloads, available from ⟨http://www.hammerofgod.com/downloads.php⟩ (accessed 2015-03-09).
[25] MacQueen, J., LeCam, L.M. and Neyman, J.: Some methods of classification and analysis of multivariate observations, *5th Berkeley Symposium on Math., Stat., and Prob.*, p.281 (1967).
[26] Kohonen, T.: Self-organized formation of topologically correct feature map, *Biol. Cybern.*, Vol.43, pp.56–69 (1982).
[27] Wu, J., Vangala, S. and Gao, L.: An Effective Architecture and Algorithm for Detecting Worms with Various Scan, *14th Annual Network and Distributed System Security Symposium* (*NDSS2004*) (2004).

Editor's Recommendation

This paper achieves to detect and prevent a sort of stealthy distributed brute force attack, being referred to as "DBF: brute force attacks with disciplined IPs", against the Remote Desktop Protocol. In recent years, cyber attacks have become sophisticated and stealthy and how to provide the countermeasure is one of big issues. The paper gives insights to readers in this research field and thus is selected as a recommended paper.

(Chief examiner of SIGSCEC Masakatsu Nishigaki)

Satomi Saito received her B.E. and M.S. degrees in 2010 (early graduated), 2012 respectively from Yokohana National University. Since 2012, she has been engaged in research and development on network security at Fujitsu Laboratories Ltd. From 2015, she is a Ph.D. candidate of Yokohama National University. She was awarded IEEJ Tokyo Branch Student Encouragement Award in 2009. She is a member of IPSJ.

Koji Maruhashi received his B.S. and M.S. degrees in 1997, 1999 respectively from Kyoto University. He received his Ph.D in engineering in 2014 from Tsukuba University. In 1999, he joined Fujitsu Ltd. Since 2002, he has been engaged in research and development on data mining and machine learning at Fujitsu Laboratories Ltd., where he is currently a senior researcher. From 2009 to 2010, he has served as visiting researcher at Carnegie Mellon University (USA).

Masahiko Takenaka received his B.E. and M.E. degrees in electronic engineering in 1990, 1992 respectively from Osaka University. He received his Ph.D. in engineering in 2009 from Tsukuba University. Since 1992, he has been engaged in research and development on cryptography, side channel analysis, network security, and cyber security at Fujitsu Laboratories Ltd. He is currently a project director. He was awarded OHM Technology Award in 2010 and IPSJ Kiyasu Special Industrial Achievement Award in 2011. He is a member of IEICE and IPSJ.

Satoru Torii received his B.E. degree in information science in 1985 from Tokyo University of Science. Since 1985, he has been engaged in research and development on network security at Fujitsu Laboratories Ltd. He is currently a research manager. He was awarded Computer Security Symposium (CSS) Paper Prize in 2004. He is a Chair of the IPSJ Special Interest Group on Computer Security (SIG-CSEC).