5-2 Framework for Countering Drive-By Download

Takahiro KASAMA, Takashi MATSUNAKA, Akira YAMADA, Ayumu KUBOTA,
Nobuyo FUJIWARA, Kazuo KAWAMORITA, and Koichiro OKADA

Drive-by Download attack (DBD attack), which exploits vulnerability in Web browser or browser's plug-in and infects user's computer with malware, have been increasing explosively. In this report, we show a framework for countering DBD attack and report experimental results of our field trial with over 1,600 participants to evaluate our framework.

1 Introduction

In the Gumblar attack that took place in 2009, the attacker manipulated compromised Web sites in a way that the users who access the sites suffer a DBD attack and get infected with malware. The malware, in turn, caused leak of FTP account information of other websites under the control of the machine. The attacker exploited the acquired FTP account information to expand the range of compromised websites, many of them belonging to major Japanese enterprises, until a large number of users suffered damage form the compromised websites.

Since the Gumblar attack was reported, many instances of DBD attacks have come to our knowledge, constituting one of major portions of malware infection incidents. One of the features that characterize DBD attacks are their passivity — i.e. the chain of events is triggered only if the user makes an access to a malicious website. Therefore, passive monitoring methods, such as darknet monitoring, do not achieve a sufficient effect, and alternative monitoring methods are required to discern the threat. The client honeypot, one of the major techniques to observe DBD attacks, simulates an environment in which a vulnerable user machine operates, and let the machine actively make access to websites on the Internet trying to trap a DBD attack in the act. However, the enormity of the number of websites on the Internet makes exhaustive inspection by the client honeypot impossible. To detect malicious website efficiently, a filtering technique is required to select out dubious URLs for further inspection. Furthermore, recently software tools called Exploit Kit have appeared which help potential attacker construct a malicious website easily. As the result, malicious websites have become increasingly ephemeral, meaning that the attackers tend to repeat the scrap-and-build process in a short cycle, in several days to several weeks, to avoid being detected. The need for quick detection has become more important in the face of this situation.

To address this situation, we, with the help from voluntary users, made an extensive deployment of web access monitoring sensors: one on each of the users' environments, which collectively constitute a macroscopic monitoring system for the web space. Based on the analysis of the web access information gathered from these sensors, we have conducted research and development to construct an anti-DBD attack framework which detects the emergence of malicious websites and compromising of benign websites. In this report, we present the overview of the anti-DBD attack framework, and then describe the results of the experiment carried out under the participation of more than 1,600 general users.

2 Overview of DBD attacks

Figure 1 shows the typical flow of events that takes place in a DBD attack. The attacker manipulates a compromised website in advance, and injects a malicious script in it to redirect the user's access to one of the attacker sites (prepared beforehand to launch an attack). Once a user makes an access to one of the compromised websites, he/she is first redirected to an intermediate site. Environment information of the user (OS, type and version of the browser, type and version of plugins, IP address, and other reference information) is scrutinized at the intermediate site, and only the users that meet the attacker's requirements are lead to the exploit server. To avoid being detected by security vendors and researchers, many intermediate sites are equipped with cloaking mechanisms,

which, based on the acquired information (IP address, referrer information, and others) determines if the access originated from a client honeypot. If it did, the access will be redirected to a regular website. When finally redirected to the exploit server — note that there may be cases in which the user is redirected through several intermediate sites — the exploit server sends the contents that are tailored to exploit the vulnerability in the user's environment. If it successfully exploited the vulnerability, the user automatically downloads and installs malware from a malware distribution server.

In recent years, software tools that assist malware preparation have been developed, such as Blackhole Exploit Kit and Angler (collectively called as Exploit Kit), and many DBD attack incidents have been reported in which they were used. Exploit kits provide types of attack codes for exploiting vulnerabilities, as well as an array of functions and tools to assist DBD attacks — i.e. obfuscation tool for exploit codes, cloaking, and web interface for control purpose. These kits relieve potential attackers from the burden of preparing the necessary tools by themselves, making setups for DBD attacks relatively easy.

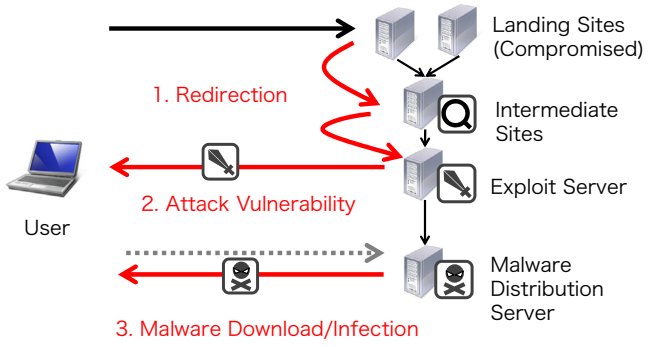

Fig. 1 Typical flow of events in DBD attack

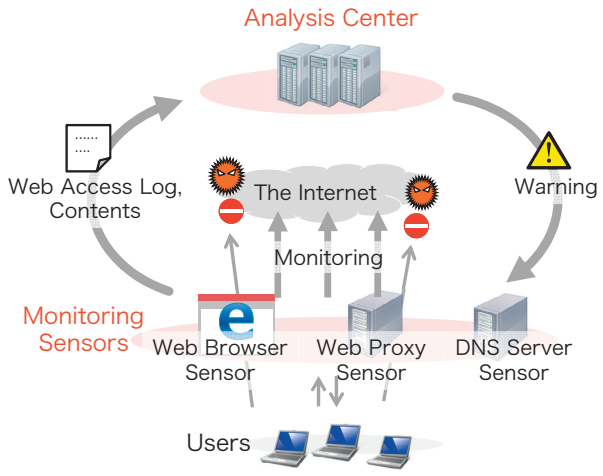

Fig. 2 Overview of the framework for countering DBD attacks

This situation has had a substantial effect on the extended damage from DBD attacks.

3 Framework for countering DBD attacks

Faced with the fact that it is a hard task to grasp the actual situation of DBD attacks, we have been conducting research and development of a framework for countering DBD attacks[1]–[3]. As described above, this project is underway in association with many general users to enable monitoring their behavior in web space from a macroscopic viewpoint. The mass of web access information from the monitoring activities allows statistical analysis for early detection of DBD attacks. The overview of this framework is shown in Fig. 2.

We have prepared three types of sensors for gathering web access information. A web browser sensor, implemented as a web browser plugin, is used as the main tool. In preparation for the situations that defy plugin installation, we have developed two other sensors: a sensor that works as a web proxy, and a DNS server sensor. The latter two sensors, however, have only limited information gathering capability in comparison with the web browser sensors operating on user machines. Therefore, our description will be focused on web browser sensor, the basic sensor, in the following sections.

3.1 Flow of processing in the framework

The web browser sensor (hereafter simply referred to as a "sensor") is plugin software for web browsers. At present, the plugin is compatible with two web browsers: Internet Explorer and Firefox. Table 1 shows the informa-

Table 1 Major information item collected by the sensor

Sensor Environment Information
Sensor ID (randomly generated at each activation)
Type and Version of Web Browser
Type and Version of Plugins
Web Browsing Information
Sensor ID (randomly generated at each activation)
Tab ID (randomly generated at each tab)
Destination URL
Destination IP address
HTTP Request/Response Header
Hash Value of Contents
Occurrence of Redirection
Occurrence of Mouse Event

tion of the environment and web browsing collected by the sensor. The sensor comes into action when the browser is activated, and creates its own ID in a randomized fashion. Then it sends information of its working environment (the sensor ID, type and version of the browser, types and versions of other plugins already in place) to the central facility for collective analysis and countermeasures (hereafter simply referred to as "analysis center"). The sensor collects web browsing information at each access from the web browser to a website and sends it to the analysis center.

Figure 3 shows the general process flow that takes place within the framework. The analysis center, upon receiving the web browsing information from the sensor, inspects its contents against the blacklist (a list of known malicious websites and contents). The blacklist contains the site URLs and hash values of contents that have been determined to be malicious by the analysis engines to be described later, as well as the already known information. In addition to the blacklist checking, heuristic engines[4][5] are also used for malignity evaluation based on such information as specific behaviors during page-to-page transition and the number of redirection steps. In case the destination website is determined as malicious, the information is transferred to the sensor, which in turn displays a warning message (dialog box) for user response. In this way, the user can disconnect the access to avoid possible attacks. If determined malicious, the analysis center sends a request, on as needed basis, to upload the web contents that have been blocked. With the permission from the user, the web contents are sent to the analysis center for detailed examination by several analysis engines[6][7]. In addition to the real-time evaluation for each web access, other engines[8][9] for the detection of malicious websites operate at regular intervals and try to detect malicious websites based on broader knowledge, such as the analysis of the pile of web browsing information gathered from a body of users and the link structure of the websites.

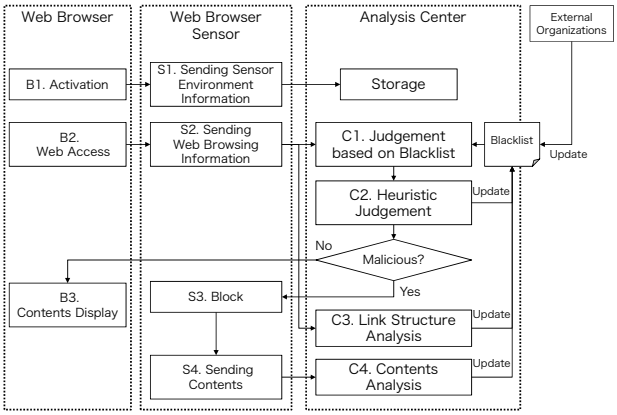

Fig. 3 Process flow inside the framework for countering DBD attacks

3.2 Considerations on user privacy

Because the access information to websites reflects the taste and adherence of each user, protection of privacy of all participating users is of utmost importance. The framework is implemented with several technical provisions to guarantee user privacy. First of all, the sensor ID of a browser sensor is generated randomly at each activation of the web browser, which effectively hinders tracking of web access history from the gathered web access information. This mechanism generates a different sensor ID every time a user restarts the browser or OS, rendering it impossible to track the same user's web browsing information for an extended period of time. In addition, several mechanisms are provided to protect user privacy: the system collects only the HTTP header information, and does not collect those types of information that require permission through dialog with the user — e.g. HTTPS communication and Cookie authentication. The user is allowed to permit/inhibit transfer of information on an item by item basis. Prior to the demonstration experiment to be described in the following section, significant efforts were made to prepare documents, terms of agreement and conditions to explain the purpose of collecting web browsing information and options available to the users. A third-party panel, consisting of learned individuals, was convened to discuss if the experiment itself and related documents are legitimate, and it concluded the validity of the procedures for the experiment.

4 Demonstration experiment with user participation

To verify the validity of this framework, a demonstration experiment was carried out, initially with the participation of around 1,000 users, from the 1st of July 2015 to the 30th of November. User recruitment continued even

Table 2 Statistics gained from the experiment

Number of Users	1,676
Total Number of Sensor IDs	49,146
Total Number of Web Accesses	4,425,689
Number of Unique Accessed URLs	2,178,381
Number of Unique Accessed FQDNs	34,195

Table 3 Web page statistics (multiply accessed web pages and their access count)

# of Unique Web Pages	2,178,381
# of Web Pages with Multiple Accesses	212,804 (9.8%)
# of Web Pages on The Alexa Top 100 Japanese Domains	56,692 (2.6%)
# of Other Web Pages	156,112 (7.2%)
Total Access Count	4,425,689
Access Count for Web Pages with Multiple Accesses	2,460,112 (55.6%)
Access Count for the Web Pages on the Alexa Top 100 Japanese Domains	668,537 (15.1%)
Access Count for Other Web Pages	1,791,575 (40.5%)

after the experiment was launched, and the total number of participating users reached 1,676, as of the 21st of October 2015. For the rest of the period, this user number was maintained. Statistics gained from the demonstration experiment are summarized in Table 2.

The number of unique URLs observed during the period of the experiment reached 2.17 million. Comparison of these URLs with the domain by domain access ranking in Japan, available from Alexa, reveals that accesses to all the top 100 domains were observed. The result indicates that, with the scale of 1,600 user participation, the accesses to major websites can be exhaustively observed, indicating the feasibility for monitoring attack incidents if one or more of these major websites are compromised by DBD attacks.

The framework assumes a statistical approach — detection of malicious websites based on access monitoring by many users, and analysis based on the accumulated data. Therefore, the power of detection is severely limited if a web page has only a single access. Table 3 shows statistics gained from the demonstration experiment: the number of web pages to which more than two accesses were observed and their access count. From this table, around 210 thousand URLs (web pages) had more than two observed accesses, and around 150 thousand of them were the web pages not included in the top 100 domains reported by Alexa. Simple scrolling on the top Alexa domain sites cannot cover all these websites. The result is indicative of the potential of our framework's approach — i.e. monitoring of general users' web accesses; it allows much wider range of web space monitoring, not limited only to the frequently visited websites. We have proposed a method to detect malicious accesses (see ref. [3]), in which an attempt is made to capture accesses to specific executable files (those defy explicit knowledge of download transitions). However, during the period of the experiment, no access was determined to be malicious in light of this method. However, eleven accesses were found to be malicious using the multi-redirection detection method — a method based on counting the number of redirection steps. Close analysis of these malicious accesses revealed that they were not directly involved in the download of malicious contents, therefore they were finally judged to be false detections. Google also provides URL monitoring tools, Safe Browsing APIs, and monitoring using these tools captured 23 potentially malicious accesses. However, close examination revealed that they did not result in actual download incidents of malwares. Thus, we concluded there was a high likelihood they were false detections.

Finally, we decided that no DBD attack occurred during the experiment's period although the possibility of false negatives remains. To resolve the situation, we consider it essential to scale up the demonstration experiment: with a larger circle of participation from the users, for the monitoring of DBD attacks and evaluation of detection engines.

5 Summary

In this report, we introduced the proposed framework for countering DBD attacks and the results of demonstration experiment conducted in collaboration with general users. The results show that, with the scale of user participation around 1,600, accesses to many websites: some of them are well known, were successfully monitored, indicating the validity of our approach to exhaustibly monitor the accesses to regular websites. In terms of malicious website detection, on the other hand, no actual DBD attacks were captured, with only a few cases of seemingly false detec-

tions. We consider it necessary to retry the experiment using more sophisticated detection engines and with a larger circle of user participation, for the monitoring of actual attacks on the spot, and for more accurate evaluation of our approach.

References

1 T. Kasama, D. Inoue, M. Eto, J. Nakazato, and K. Nakao, "A Framework for Countering Drive-by Download Attacks," In Proceedings of the IPSJ Computer Security Symposium 2011 (CSS2011), October 2011.

2 T. Matsunaka, J. Urakawa, and A. Kubota, "Detecting and Preventing Drive-by Download Attack via Participative Monitoring of the Web," In Proceedings of the 8th Asia Joint Conference on Information Security (AsiaJCIS 2013), July 2013.

3 T. Matsunaka, J. Urakawa, A. Nakarai, A. Kubota, K. Kawamorita, Y. Hoshizawa, T. Kasama, M. Eto, D. Inoue, and K. Nakao, "FCDBD: Framework for Countering Drive-by Download," The 9th International Workshop on Security (IWSEC2014), poster session, August 2014.

4 T. Kasama, M. Kamizono, and D. Inoue, "Drive-by-Download Attack Detection based on Characteristics of Exploit Kit," In Proceedings of the Anti Malware Engineering Workshop 2013 (MWS2013), October 2013.

5 T. Matsunaka, A. Kubota, and T. Kasama, "An Approach to Detect Drive-by Download by Observing the Web Page Transition Behaviors," In Proceedings of the 9th Asia Joint Conference on Information Security (AsiaJCIS 2014), September 2014.

6 M. Nishida, Y. Hoshizawa, T. Kasama, M. Eto, D. Inoue, and K. Nakao, "Obfuscated Malicious JavaScript Detection using Machine Learning with Character Frequency," IPSJ SIG Notes 2014-CSEC-64(21), March 2014.

7 M. Kamizono, K. Iwamoto, T. Kasama, M. Eto, D. Inoue, and K. Nakao, "Development of an Environment-independent Dynamic Analysis System for Document Malware," IEICE technical report. Information and communication system security, vol.114, no.71, June 2014.

8 T. Matsunaka, A. Nakarai, J. Urakawa, and A. Kubota, "A Consideration of Detecting Compromised Web Sites by Analyzing Web Link Structures in the Framework for Combating Drive-by Download Attacks," In Proceedings of the 31st Symposium on Cryptography and Information Security (SCIS 2014), January 2014.

9 T. Kasama, M. Eto, M. Kamizono, and D. Inoue, "Malicious Web Site Detection Based on Redirection Control using Client Environment," IEICE technical report. Information and communication system securit, vol.114, no.71, June 2014.

Takahiro KASAMA, Ph.D.
Researcher, Cybersecurity Laboratory,
Cybersecurity Research Institute
Cybersecurity

Takashi MATSUNAKA
KDDI CORPORATION

Akira YAMADA
KDDI Research, Inc.

Ayumu KUBOTA
KDDI Research, Inc.

Nobuyo FUJIWARA
SecureBrain Corporation

Kazuo KAWAMORITA
SecureBrain Corporation

Koichiro OKADA
SecureBrain Corporation