BLINDINGCAN - Malware Used by Lazarus - - JPCERT/CC Eyes

By 朝長 秀誠 (Shusei Tomonaga)

Published: 2020-09-28 · Archived: 2026-04-05 12:39:38 UTC

- Lazarus

In the previous article, we introduced one type of malware that Lazarus (also known as Hidden Cobra) uses after network intrusion. It is confirmed that this attack group uses multiple types of malware including BLINDINGCAN, which CISA recently introduced in its report [1].
This article summarises the result of our analysis on BLINDINGCAN.

BLINDINGCAN overview

The malware runs when a loader loads a DLL file. Figure 1 shows the flow of events until BLINDINGCAN runs. JPCERT/CC has confirmed that the DLL file is encoded in some samples (which requires decoding by the loader before execution).

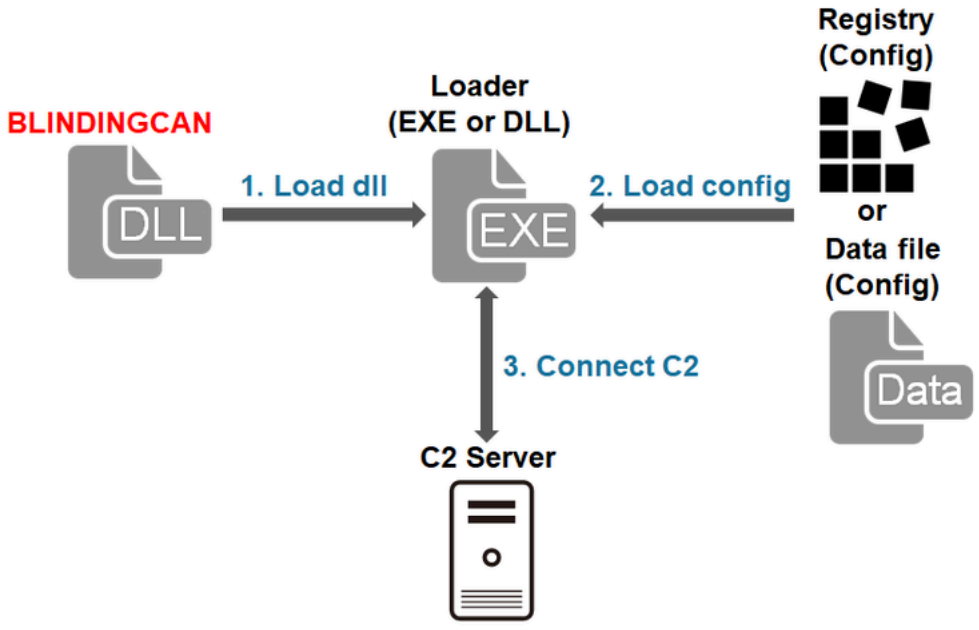


Figure 1: BLINDINGCAN behaviour

BLINDINGCAN shares some features with the aforementioned malware including its function and communication encoding algorithm. The following sections will explain its configuration and communication protocol.

Configuration

The configuration of BLINDINGCAN(size: 0xA84) is stored in one of the following locations:

- Hardcoded in the malware itself
- Stored in a registry entry
- Saved as a file

In case it is saved as a file, it is stored in the same folder where BLINDINGCAN is located. We have confirmed that the following directory is used if the configuration is stored in a registry entry.

```
Key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion
Value: "SM_Dev16[numeric string]"
```

The configuration is encrypted using either XOR encoding, AES or RC4. The encryption key is either fixed or generated based on the environment of the infected device. JPCERT/CC has confirmed the following patterns of encryption keys:

- [File name][Export function name][Service name]
- [CPUID][Computer name][Processor name][Physical memory size]

Figure 2 shows an example of decoded configuration. This includes proxy information as well as C&C server information. (Please see Appendix A for details.)

```
00000000  67 2d 51 44 1d e5 00 3c  05 00 00 00 68 74 74 70  |g-QD...<....http|
00000010  73 3a 2f 2f 77 77 77 2e  61 75 74 6f 6d 65 72 63  |s://www.automerc|
00000020  61 64 6f 2e 63 6f 2e 63  72 2f 65 6d 70 6c 65 6f  |ado.co.cr/empleo|
00000030  2f 63 73 73 2f 6d 61 69  6e 2e 6a 73 70 00 00 00  |/css/main.jsp...|
00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000110  68 74 74 70 73 3a 2f 2f  77 77 77 2e 61 75 74 6f  |https://www.auto|
00000120  6d 65 72 63 61 64 6f 2e  63 6f 2e 63 72 2f 65 6d  |mercado.co.cr/em|
00000130  70 6c 65 6f 2f 63 73 73  2f 6d 61 69 6e 2e 6a 73  |pleo/css/main.js|
00000140  70 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |p...............|
00000150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000210  00 00 00 00 68 74 74 70  73 3a 2f 2f 77 77 77 2e  |....https://www.|
00000220  61 75 74 6f 6d 65 72 63  61 64 6f 2e 63 6f 2e 63  |automercado.co.c|
00000230  72 2f 65 6d 70 6c 65 6f  2f 63 73 73 2f 6d 61 69  |r/empleo/css/mai|
00000240  6e 2e 6a 73 70 00 00 00  00 00 00 00 00 00 00 00  |n.jsp...........|
00000250  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000310  00 00 00 00 00 00 00 00  68 74 74 70 73 3a 2f 2f  |........https://|
00000320  77 77 77 2e 63 75 72 69  6f 66 69 72 65 6e 7a 65  |www.curiofirenze|
00000330  2e 63 6f 6d 2f 69 6e 63  6c 75 64 65 2f 69 6e 63  |.com/include/inc|
00000340  2d 73 69 74 65 2e 61 73  70 00 00 00 00 00 00 00  |-site.asp.......|
00000350  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000410  00 00 00 00 00 00 00 00  00 00 00 00 68 74 74 70  |............http|
00000420  73 3a 2f 2f 77 77 77 2e  6e 65 2d 62 61 2e 6f 72  |s://www.ne-ba.or|
00000430  67 2f 66 69 6c 65 73 2f  6e 65 77 73 2f 74 68 75  |g/files/news/thu|
00000440  6d 62 73 2f 74 68 75 6d  62 73 2e 61 73 70 00 00  |mbs/thumbs.asp..|
00000450  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000520  01 00 00 00 0a 0a 0a 0a  30 30 30 00 00 00 00 00  |........0/0.....|
00000530  00 00 00 00 00 00 00 00  00 00 3c 00 00 00 00 00  |..........<.....|
00000540  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000660  00 00 00 00 06 00 00 00  00 00 00 00 00 00 00 00  |................|
00000670  00 00 00 00 00 00 00 00  00 00 00 00 63 00 3a 00  |............c.:.|
00000680  5c 00 77 00 69 00 6e 00  64 00 6f 00 77 00 73 00  |¥.w.i.n.d.o.w.s.|
00000690  5c 00 73 00 79 00 73 00  74 00 65 00 6d 00 33 00  |¥.s.y.s.t.e.m.3.|
000006a0  32 00 5c 00 63 00 6d 00  64 00 2e 00 65 00 78 00  |2.¥.c.m.d...e.x.|
000006b0  65 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |e...............|
000006c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000880  00 00 00 00 25 00 74 00  65 00 6d 00 70 00 25 00  |....%.t.e.m.p.%.|
00000890  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000a80  00 00 00 00 00 00 00 00  00 00 00 00              |............|
```

Figure 2: Configuration example

Obfuscation

Some part of code in BLINDINGCAN is obfuscated using RC4. Figure 3 is an example of obfuscated code. The RC4 encryption key is hardcoded in the sample itself.

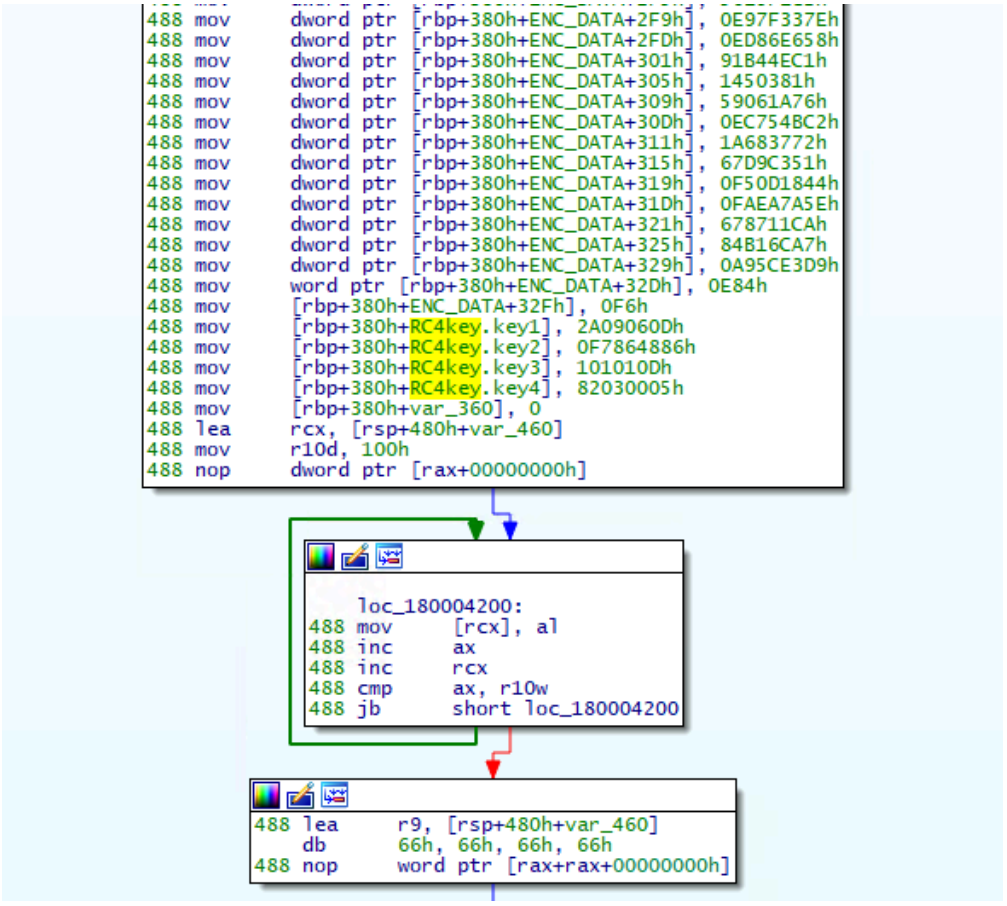


Figure 3: Obfuscation

Communication with C&C server

Below is an example HTTP POST request data that BLINDINGCAN sends in the beginning.

```
POST /[PATH] HTTP/1.1
Connection: Keep-Alive
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) Chrome/28.0.1500.95 Safari/537.36
Host: [Server]
Content-Length: [Length]

id=d3Ztd3lod2t0Tqf42ux9uv3FGH+Y3oAc2w==&bbs=HA==&tbl=hzE4dlKcRq3gokgAGeMQug== &bbs_form=4GQAAA==
```

The format of the data is as follows. (All the values except for the RC4 key is RC4-encrypted and Base64-encoded.) The param2 in the first HTTP POST request is the encoded value of a string "T1B7D95256A2001E".

```
id=[RC4 key][param1:param2:param3]&[param1]=[Random value (between 1000 and 10000)]&[param2]=["T1B7D
```

The parameters in the POST data (param1, param2, param3) are randomly selected from the below:

```
boardid,bbsNo,strBoardID,userid,bbs,filename,code,pid,seqNo,ReportID,v,PageNumber,num,view,read,acti
```

The RC4 encryption used here is different from the regular one. It has a process to shift the key stream for C00h times. The following is the RC4 encryption process written in Python. It does not apply to param3, which uses the regular RC4.

```
def custom_rc4(data, key):
    x = 0
    box = list(range(256))
    for i in range(256):
        x = (x + int(box[i]) + int(key[i % len(key)])) % 256
        box[i], box[x] = box[x], box[i]

    x = 0
    for i in range(0xC00):
        i = i + 1
        x = (x + int(box[i % 256])) % 256
        wow_x = x
        box[i % 256], box[x] = box[x], box[i % 256]
        wow_y = i % 256

    x = wow_y
    y = wow_x
    out = []
    for char in data:
        x = (x + 1) % 256
        y = (y + box[x]) % 256
        box[x], box[y] = box[y], box[x]
        out.append(chr(char ^ box[(box[x] + box[y]) % 256]))

    return ''.join(out)
```

Figure 4 is the flow of communication from the beginning of its communication with a C&C server until receiving commands.

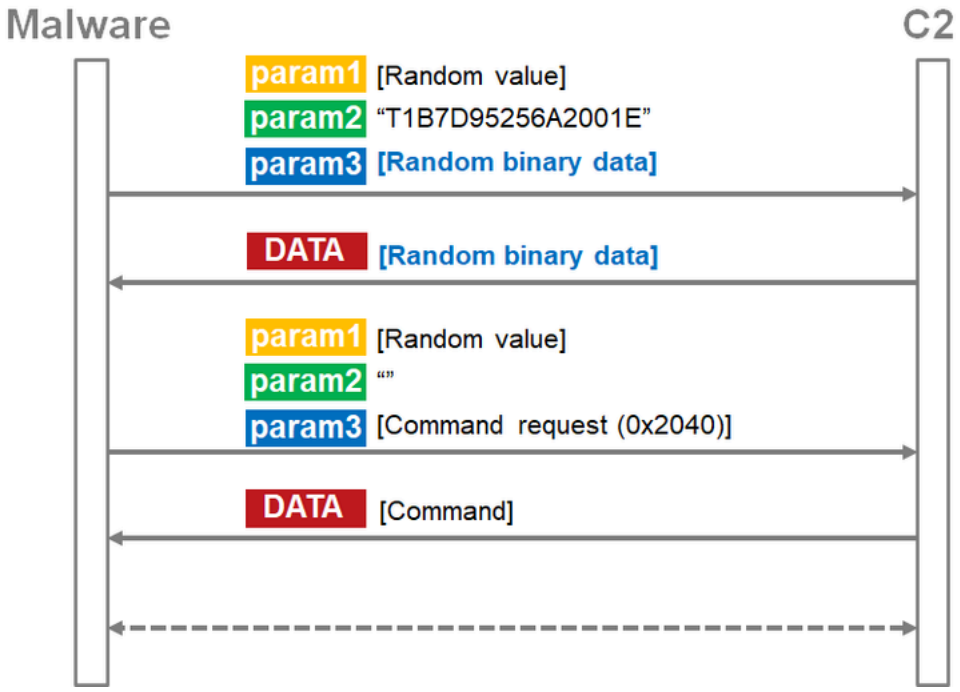


Figure 4: BLINDINGCAN communication flow

If a Base64-encoded value of param3 ([Random binary data] in Figure 4) is received from the server as a response to the first request, the malware sends another request.
The next data is sent with empty param2 and a command request (Command request (0x2040) in Figure 4) in param3. (Please see Appendix B for the details of param3 format.) The data in param3 is XOR-encoded, RC4-encrypted and then Base64-encoded.
After that BLINDINGCAN receives a command from a C&C server. (The format of the response data is the same as param3. Please see Appendix B). The response data is also XOR-encoded, RC4-encrypted and Base64-encoded. The only difference is that the "+" sign is replaced by a space.

Commands

BLINDINGCAN performs multiple functions including the following. (Please see Appendix C for details.)

- Operation on files (create a list, delete, move, modify timestamp, copy)
- Operation on processes (create a list, execute, kill)
- Upload/download files
- Obtain disk information
- Obtain a list of services
- Execute arbitrary shell command

In closing

We have introduced two kinds of malware used by Lazarus so far. However, they are known to use other types of malware as well. We will provide an update if we observe any new kind of malware.
The C&C server information of the samples mentioned in the article are listed in Appendix D. Please make sure that none of your device is communicating with these hosts.

Shusei Tomonaga
(Translated by Yukako Uchida)

Reference

[1] CISA: Malware Analysis Report (AR20-232A)
https://us-cert.cisa.gov/ncas/analysis-reports/ar20-232a

Appendix A: Configuration

Table A: List of configurations

Offset	Description	Remarks
0x000	Number of C&C servers	Up to 5
0x004	C&C server 1	
0x108	C&C server 2	
0x20C	C&C server 3	
0x310	C&C server 4	
0x414	C&C server 5	
0x518	Flag for proxy	
0x51C	Proxy IP address	
0x520	Proxy port number	
0x522	Flag for C&C reply	
0x526	Flag for drive information	
0x52A	Flag for session information	
0x52E	Save configuration	
0x532	Communication interval	
0x534	Start time	
0x53C	seed	Used when generating random data to send
0x59C	File name	File obtained by the command "0x2039"
0x5FC	Unknown	
0x65C	Unknown	

0x660	Not assigned	
0x674	Execute process name	Contains "c:¥windows¥system32¥cmd.exe"
0x87C	Temp folder	Command execution result is stored

Appendix B: Contents of data exchanged

Table B: Data format of param3 and response data

Offset	Length	Contents
0x00	2	Not assigned
0x02	2	Command
0x04	4	Not assigned
0x08	4	Parameter length
0x0C	-	Parameter (Base64 + RC4)

Appendix C: Commands

Table C: List of commands

Value	Contents
0x2009	Get system information
0x2010	Get drive information
0x2011	Get directory list
0x2012	Execute command (Regular output)
0x2013	Upload file (compressed in zlib)
0x2014	Download file
0x2015	Execute process
0x2016	Execute process (CreateProcessAsUser)
0x2019	Get process list
0x2020	Kill process
0x2021	Delete file (sdelete)
0x2022	Check connection

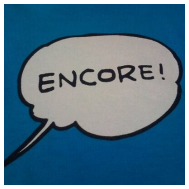
0x2023	Change current directory
0x2027	Modify file creation time
0x2028	Change communication interval with C&C server
0x2029	End session with C&C server
0x2030	Uninstall
0x2031	Get configuration
0x2032	Overwrite configuration
0x2033	Get directory information
0x2034	Get drive available space
0x2037	-
0x2038	Sleep
0x2039	Get file name
0x2046	Write file
0x2048	Copy file
0x2049	Move file
0x2050	Delete file

Appendix D: C&C server

- https://www.automercado.co.cr/empleo/css/main.jsp
- https://www.curiofirenze.com/include/inc-site.asp
- https://www.ne-ba.org/files/news/thumbs/thumbs.asp
- https://www.sanlorenzoyacht.com/newsl/include/inc-map.asp

Appendix E: Sample hash value

- 8db272ea1100996a8a0ed0da304610964dc8ca576aa114391d1be9d4c5dab02e
- 58027c80c6502327863ddca28c31d352e5707f5903340b9e6ccc0997fcb9631d



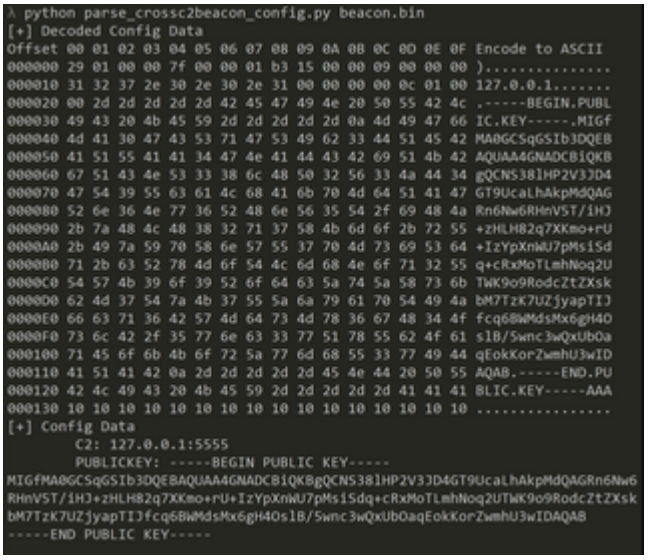
朝長 秀誠 (Shusei Tomonaga)

Since December 2012, he has been engaged in malware analysis and forensics investigation, and is especially involved in analyzing incidents of targeted attacks. Prior to joining JPCERT/CC, he was engaged in security monitoring and analysis operations at a foreign-affiliated IT vendor. He presented at CODE BLUE, BsidesLV, BlackHat USA Arsenal, Botconf, PacSec and FIRST Conference. JSAC organizer.

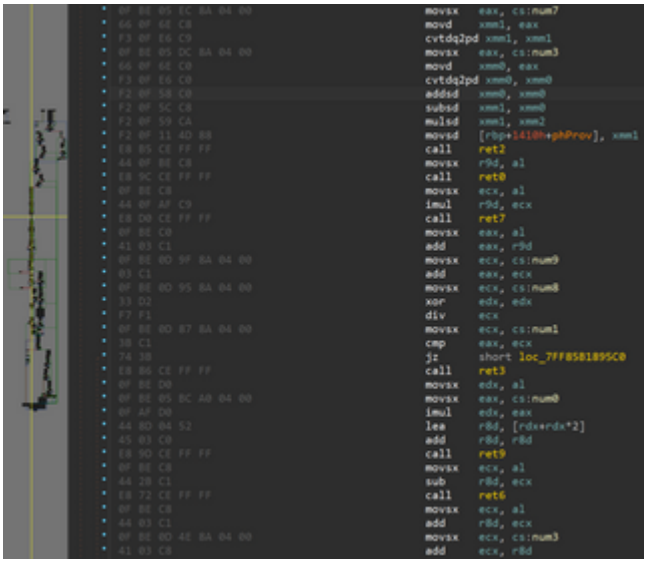
Related articles

- 

  Update on Attacks by Threat Group APT-C-60

- 

  CrossC2 Expanding Cobalt Strike Beacon to Cross-Platform Attacks

- 

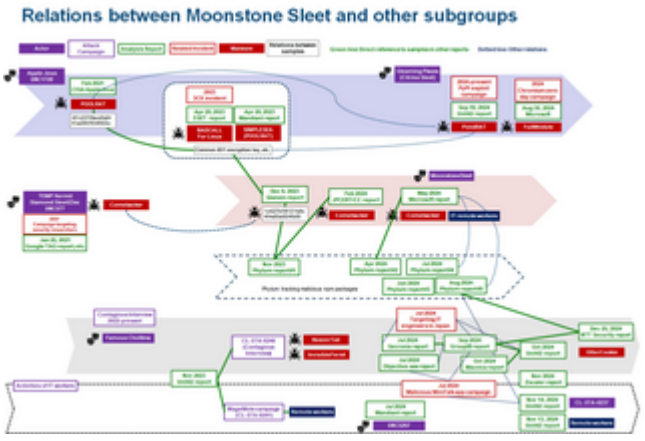
  Malware Identified in Attacks Exploiting Ivanti Connect Secure Vulnerabilities

- 

```
__int64 __fastcall mal_decode(__int64 encbuf, int bufsize)
{
  __int64 j_1; // rax
  int i; // [rsp+18h] [rbp-Ch]

  if ( encbuf )
  {
    for ( i = 0; ; ++i )
    {
      j_1 = (unsigned int)i;
      if ( i >= bufsize )
        break;
      *(_BYTE *)(encbuf + i) ^= Key1to7[i % 7];
    }
  }
  return j_1;
}
```

  DslogdRAT Malware Installed in Ivanti Connect Secure

- 

  Tempted to Classifying APT Actors: Practical Challenges of Attribution in the Case of Lazarus's Subgroup

---

Source: https://blogs.jpcert.or.jp/en/2020/09/BLINDINGCAN.html