Persistent Threats from the Kimsuky Group Using RDP Wrapper - ASEC

By ATCP

Published: 2025-02-03 · Archived: 2026-04-05 19:48:16 UTC

AhnLab SEcurity intelligence Center (ASEC) has previously analyzed cases of attacks by the Kimsuky group, which utilized the PebbleDash backdoor and their custom-made RDP Wrapper. The Kimsuky group has been continuously launching attacks of the same type, and this post will cover additional malware that have been identified.

1. Overview

Threat actors are distributing a shortcut file (*.LNK) containing a malicious command through spear-phishing attacks. The fact that the file names include names and company names suggests that the threat actors may be gathering information on specific targets.

The shortcut malware is disguised as a document file with an Office document icon such as PDF, Excel, or Word. When this file is executed, PowerShell or Mshta is run to download and execute additional payloads from external sources. The malware that is ultimately executed to control the infected system are PebbleDash and RDP Wrapper. The threat actor has recently created and distributed PebbleDash and RDP Wrapper, but there are no significant differences from previous attack cases.

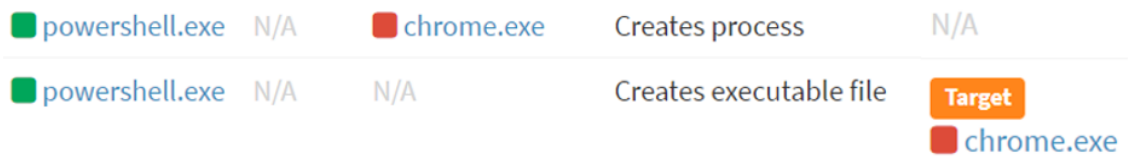

Figure 1. A PowerShell process installing the PebbleDash dropper

For reference, RDP Wrapper is an open-source utility that supports the remote desktop feature. Since Windows operating systems do not support remote desktop in all versions, RDP Wrapper can be installed in such environments to activate remote desktop. The threat actor is using RDP Wrapper that they created themselves. It is suspected that they are creating Export functions in various ways to bypass file detection.

Ordinal	RVA	Name RVA	Name
0001	00005280	0001B4FC	ServiceMain
0002	00001210	0001B508	StringFinder
0003	000052E0	0001B515	SvchostPushServiceGlobals
0004	000011E0	0001B52F	WindowsUpdate

Ordinal	RVA	Name RVA	Name
0001	00001000	0001F19C	GetWindowTextInfo
0002	00001ED0	0001F1AE	ServiceMain
0003	00001F10	0001F1BA	SvchostPushServiceGlobals
0004	00011000	0001F1D4	WinLoadString

Ordinal	RVA	Name RVA	Name
0001	00010000	00021586	CoInsiamtal
0002	00011000	00021592	OpenPipeStr
0003	000151D0	0002159E	ServiceMain
0004	00015220	000215AA	SvchostPushServiceGlobals
0005	00010090	000215C4	TpSetWait

Figure 2. Export functions of the self-developed RDP Wrapper

Threat actors can control the infected system using PebbleDash and RDP Wrapper, but they also utilize a variety of other malware, such as Proxy, KeyLogger, and information-stealing malware. This post will cover the types identified since the last post.

2. Proxy

Even if the RDP service is activated and a user account is added, external access to the infected system is not possible if it is located in a private network. To address this issue, threat actors install proxy malware that serves as an intermediary between the infected system and an external network, allowing them to access the system via RDP.

In the previous attacks, three main types of proxy tools were used. The first type is characterized by creating a mutex named “MYLPROJECT” and was identified along with a launcher. The launcher reads a configuration file located in a hard-coded path such as “C:\Programdata\USOShared2\version.ini” and uses this information to execute the proxy tool located in a specific path. The second type of proxy tool is characterized by creating a mutex named “LPROXYMUTEX” and is otherwise the same as a typical proxy. The last type is a Go language-based revsocks tool that is publicly available on GitHub.

The recently identified proxy tools use the following mutexes and receive addresses as arguments to operate.

```
ConsoleWindow = GetConsoleWindow();
ShowWindow(ConsoleWindow, 0);
MutexA = CreateMutexA(0LL, 1, "8iwUDMK0kskwUK14WEKAI9NDMHS474KAEJKN6QDIWDAP8");
if ( !((MutexA == 0LL) | (GetLastError() == 183)) )
```

```
    WSAGetLastError();
    v3 = "[Client] Send to Local socket error. errcode : %d";
    goto LABEL_8;
  }
  if ( !dword_140023B78 )
    goto LABEL_9;
}
WSAGetLastError();
v3 = "[Client] Recv from Server socket error. errcode : %d";
```

Figure 3. A proxy tool similar to the previous type

3. KeyLogger

The Kimsuky group uses a PowerShell script to perform keylogging and also installs keyloggers in executable file format. In previous cases, the group mainly stored user keystrokes in the "%LOCALAPPDATA%\CursorCach.tmp" and "%LOCALAPPDATA%\CursorCache.db" paths. However, the recently identified types are characterized by storing the data in the "C:\Programdata\joeLog.txt" and "C:\Programdata\jLog.txt" paths.

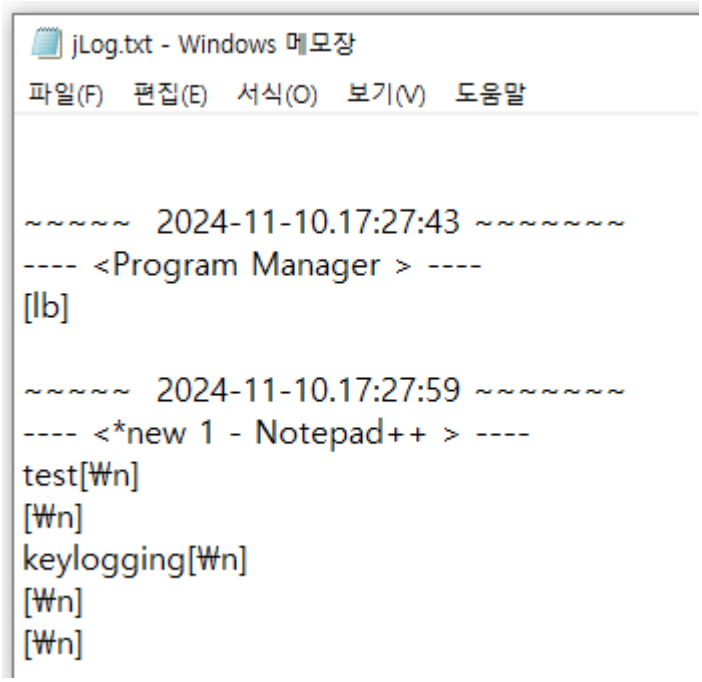

Figure 4. Keylogging file

4. Theft of Web Browser Information (forceCopy)

In the previous cases, Infostealer malware were used to steal user credentials stored in Chromium-based web browsers and Internet Explorer. Recently, additional cases of the same type of malware have been identified.

The Kimsuky group used a tool that extracts only the key value from the “Local State” file instead of directly stealing credentials stored in the web browser. This is presumed to be for bypassing security products, and the extracted key is used later in the process of stealing credentials stored in the web browser.

The recently discovered type is installed under the name “forceCopy” and is used to copy files. It receives the path of the file to be copied as the first argument and the path where the file will be saved as the second argument. A characteristic of this malware is that it uses the NTFS Parser library to read files instead of APIs like ReadFile().

```
.data:0000000140026F18   00000010     C      .?AVCAttrBase@@
.data:0000000140026D98   00000017     C      .?AVCAttrNonResident@@
.data:0000000140026DC0   00000014     C      .?AVCAttrResident@@
.data:0000000140026D48   00000015     C      .?AVCAttr_FileName@@
.data:0000000140026C80   00000017     C      .?AVCAttr_IndexAlloc@@
.data:0000000140026CA8   00000016     C      .?AVCAttr_IndexRoot@@
.data:0000000140026D70   00000014     C      .?AVCAttr_StdInfo@@
.data:0000000140026D20   00000014     C      .?AVCAttr_VolInfo@@
.data:0000000140026CF8   00000014     C      .?AVCAttr_VolName@@
.data:0000000140026ED0   00000010     C      .?AVCFileName@@
.data:0000000140026E80   00000012     C      .?AVCFileRecord@@
.data:0000000140026CD0   00000012     C      .?AVCIndexBlock@@
.data:0000000140026EF0   00000012     C      .?AVCIndexEntry@@
.data:0000000140026EA8   00000012     C      .?AVCNTFSVolume@@
```

Figure 5. NTFS Parser library included in the malware

All of the paths where the malware is installed are web browser installation paths. It is assumed that the threat actor is attempting to bypass restrictions in a specific environment and steal the configuration files of the web browsers where credentials are stored. This may also be to bypass security products, similar to past cases.

5. Loader, Injector

The difference from previous cases is the identification of Injector and Loader malware. While the malware that ultimately operates in the memory has not been identified, the Loader loads a file from the “%SystemDirectory%\wbemback.dat” path into the memory, and the Injector receives information such as the target process for injection as an argument to operate.

In addition to malware in the form of executable files, ReflectiveLoader has also been identified among PowerShell scripts. It is obfuscated, but it is an open-source PowerShell script called “Invoke-ReflectivePEInjection.ps1”. It is installed along with other PowerShell script malware in the “%ALLUSERSPROFILE%\USOShared\Prosd\” directory.

```
            &("{1}{3}{2}{0}" -f'e','Set','l','-Variab') -Name ("{0}{1}" -f 'Succes','s') -Value
            if (${sU`C`CesS} -eq ${f`Al`se})
            {
                &("{0}{1}{3}{2}" -f 'W','rit','ng','e-Warni') ((("{13}{10}{11}{8}{0}{2}{14}{15}{
            }
        }
    }
    Main
}

Function Main
{
    [Byte[]]$bytes = [System.IO.File]::ReadAllBytes($ZzPath);
    $length = $bytes.Length;
    $bytes[10] = 0x1f;
    [byte[]]$PEBytes = MCknsiuheg545y ($bytes[10..($length-1)]);

    $e_magic = ($PEBytes[0..1] | % {[Char] $_}) -join ''

    if ($e_magic -ne 'MZ')
    {
        throw 'PE is not a valid PE file.'
```

Figure 6. ReflectiveLoader PowerShell script

6. Conclusion

In 2024, the attack methods of the Kimsuky group changed. While the use of LNK malware in spear-phishing attacks during the initial breach remained the same, the group began to increasingly use tools such as RDP Wrapper and Proxy to remotely control the infected systems instead of installing backdoors.

The Kimsuky threat group is continuously launching spear phishing attacks against Korean users. They mainly distribute malware disguised as a document file attached to an email, and if a user executes this file, threat actors can take control of the system. Users must carefully check the sender of the email and refrain from opening files from unknown sources. Users should also apply the latest patches for programs such as their OS and web browsers, and update AhnLab V3 to the latest version so that malware infection can be prevented.

File Detection

Backdoor/Win.PebbleDash.C5719351 (2025.01.20.02)
Trojan/Win.Rdpwrap.C5704469 (2024.12.10.02)
Trojan/Win.Rdpwrap.C5708551 (2024.12.21.00)
Trojan/Win.Rdpwrap.C5710893 (2024.12.27.00)
Trojan/Win.Rdpwrap.C5716647 (2025.01.12.03)
Trojan/Win.Rdpwrap.C5719870 (2025.01.21.03)
Trojan/Win.Rdpwrap.C5720371 (2025.01.23.00)
Trojan/Win.KeyLogger.C5687683 (2024.10.27.03)
Trojan/Win.KeyLogger.C5705213 (2024.12.12.01)
Trojan/Win.KeyLogger.C5705571 (2024.12.13.00)
Trojan/Win.Injecter.C5705214 (2024.12.12.01)
Trojan/Win.UACMe.C5705215 (2024.12.12.01)
Trojan/Win.Loader.C5716648 (2025.01.12.03)

Infostealer/Win.Browser.R641029 (2024.03.23.00)
Malware/Gen.Generic.C2950389 (2019.01.22.01)
Trojan/Win.Agent.C5687684 (2024.10.27.03)
Trojan/PowerShell.Loader (2025.01.31.02)
Trojan/PowerShell.Launcher (2025.01.31.02)
Trojan/PowerShell.KeyLogger (2025.01.31.02)

MD5

04e5f813da28b5975d0b6445f687bc48

26d96d40e4c8aed03d80740e1d5a4559

2ea71ff410088bbe79f28e7588a6fb47

3211ef223177310021e174c928f96bab

5565b337bfba78970b73ae65b95f2c4f

Additional IOCs are available on AhnLab TIP.

IP

216[.]219[.]87[.]41

74[.]50[.]94[.]175

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/86098/