SUBEX

Viking Virus

Date: 21/07/2021

Shyava Tripathi

Albeit a weathered and decelerated infection vector, file infectors harnessing the capabilities of hybrid payloads and crossbreeding propagation strategies are yet pervasive in nature. Viking, a file infector prepending virus subsuming backdoor & information theft routines has been able to withstand the test of time.

Summary

Viking infects executable files on all available fixed hard drives and infects them by prepending its malicious code before the original file's body. However, the virus infects only Windows NT-based systems.

The infection starts by the virus creating a backup copy of itself in the current directory, then executing the original file followed by deleting it & renaming the dropped copy to the original filename. This is accomplished by making use of a temporary batch file dropped in the system's temporary folder.

The executable drops two other embedded executables in the system's 'Windows' folder, which in turn drops another DLL in the same directory. The DLL runs a keylogger and is also capable of downloading and running files from a website. The virus then scans all folders and subfolders for executables to infect and drops a file, '_desktop.ini' in every infected folder which contains the current date. Upon execution of an infected file, the virus's infection routine infects other files, although, it removes itself from the file being executed, leaving it clean.

Analysis

- **Execution**

The files are packed with a generic packer & contain a section '.Upack' pertaining to the obfuscation.

The infection is initiated by the virus dropping the following three files, one in the system's 'Temp' directory & the rest in the system's 'Windows' directory.

- C:\Users\<>\AppData\Local\Temp\$$a1C1B.bat
- C:\Windows\Logo1_.exe
- C:\Windows\rundl132.exe

Upon execution of the sample, the virus creates a backup copy of itself in the current directory by the name [filename].exe.exe, then executing the original file, [filename].exe followed by deleting it & renaming the dropped copy to the original filename. The batch file, '$$a1C1B.bat' (Figure 1) is used to carry out the operations.

```
:try1
Del "C:\Users\cuckoo2\AppData\Local\Temp\2694e399ed931a811067f05499aed0e14184d369ba50476262d8f3d42ff43fe4.exe"
if exist "
    C:\Users\cuckoo2\AppData\Local\Temp\2694e399ed931a811067f05499aed0e14184d369ba50476262d8f3d42ff43fe4.exe"
    goto try1
ren "
    C:\Users\cuckoo2\AppData\Local\Temp\2694e399ed931a811067f05499aed0e14184d369ba50476262d8f3d42ff43fe4.exe.exe
    " "2694e399ed931a811067f05499aed0e14184d369ba50476262d8f3d42ff43fe4.exe"
if exist "
    C:\Users\cuckoo2\AppData\Local\Temp\2694e399ed931a811067f05499aed0e14184d369ba50476262d8f3d42ff43fe4.exe.exe
    " goto try2
"C:\Users\cuckoo2\AppData\Local\Temp\2694e399ed931a811067f05499aed0e14184d369ba50476262d8f3d42ff43fe4.exe"
:try2
del "C:\Users\cuckoo2\AppData\Local\Temp\$$aC090.bat"
```

Figure 1: Batch File

The two other executable files dropped in the 'Windows' folder, 'Logo1_.exe' & 'rundl132.exe' are embedded in the main sample. These two executables run code which help in achieving persistence, infecting files further & announcing the infection to an attacker controlled server. 'Rundl132.exe', in turn, drops another DLL component of the infector, 'vDLL.DLL' which runs a

keylogger and is capable of downloading and running files from a website. The DLL is injected into the 'Explorer.exe' process for execution (Figure 2).

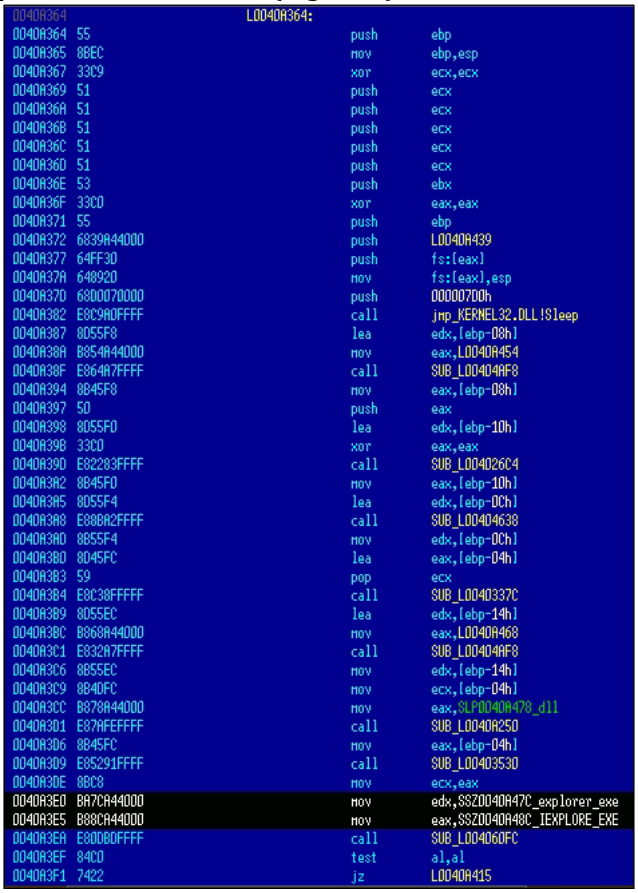


Figure 2: Injection in explorer.exe

The virus then scans all the folders and subfolders for executables (Figure 4) to infect, and drops a file, '_desktop.ini' in every infected folder (Figure 3) which contains the current date in YYYY/MM/DD format.

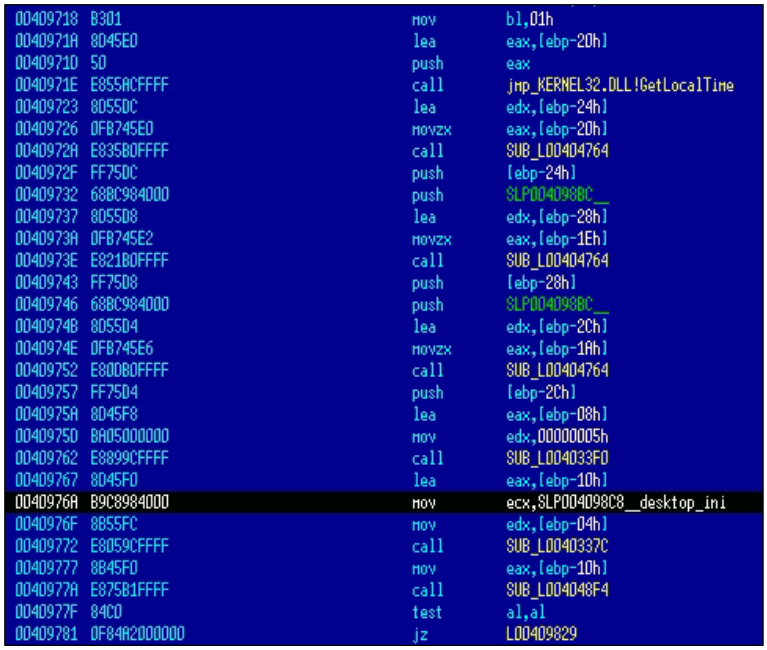
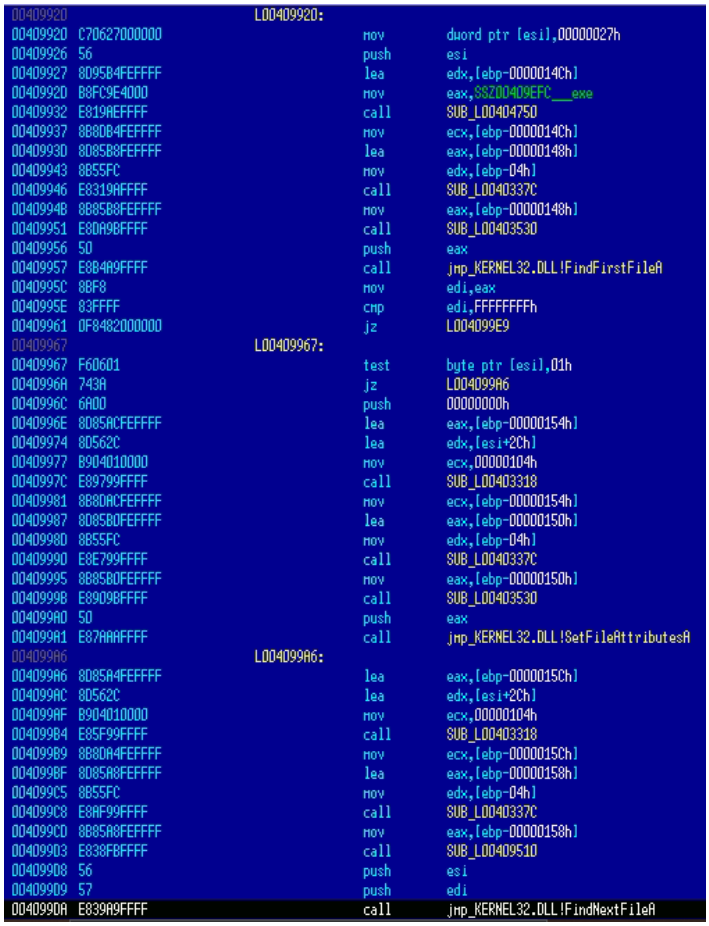


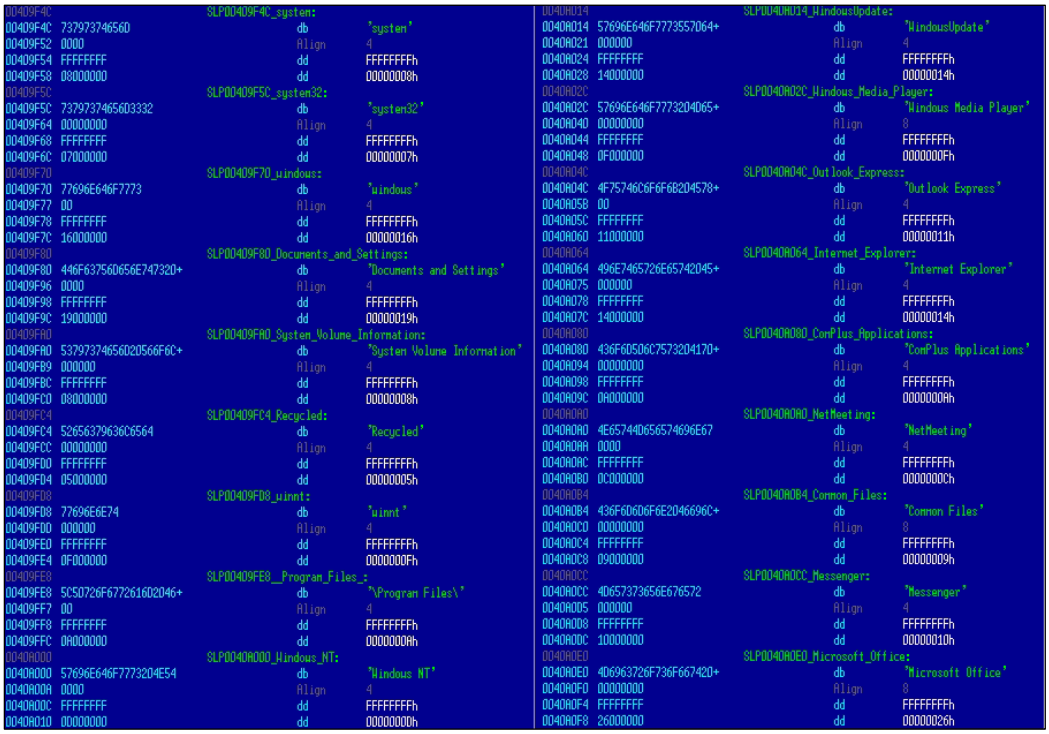
Figure 3: Function dropping desktop.ini



```
00409920                        L00409920:
00409920 C70627000000                   mov     dword ptr [esi],00000027h
00409926 56                             push    esi
00409927 8D95B4FEFFFF                   lea     edx,[ebp-0000014Ch]
0040992D B8FC9E4000                     mov     eax,SSZ00409EFC___exe
00409932 E819AEFFFF                     call    SUB_L00404750
00409937 8B8DB4FEFFFF                   mov     ecx,[ebp-0000014Ch]
0040993D 8D85B8FEFFFF                   lea     eax,[ebp-00000148h]
00409943 8B55FC                         mov     edx,[ebp-04h]
00409946 E8319AFFFF                     call    SUB_L0040337C
0040994B 8B85B8FEFFFF                   mov     eax,[ebp-00000148h]
00409951 E8DA9BFFFF                     call    SUB_L00403530
00409956 50                             push    eax
00409957 E8B4A9FFFF                     call    jmp_KERNEL32.DLL!FindFirstFileA
0040995C 8BF8                           mov     edi,eax
0040995E 83FFFF                         cmp     edi,FFFFFFFFh
00409961 0F8482000000                   jz      L004099E9
00409967                        L00409967:
00409967 F60601                         test    byte ptr [esi],01h
0040996A 743A                           jz      L004099A6
0040996C 6A00                           push    00000000h
0040996E 8D85ACFEFFFF                   lea     eax,[ebp-00000154h]
00409974 8D562C                         lea     edx,[esi+2Ch]
00409977 B904010000                     mov     ecx,00000104h
0040997C E89799FFFF                     call    SUB_L00403318
00409981 8B8DACFEFFFF                   mov     ecx,[ebp-00000154h]
00409987 8D85B0FEFFFF                   lea     eax,[ebp-00000150h]
0040998D 8B55FC                         mov     edx,[ebp-04h]
00409990 E8E799FFFF                     call    SUB_L0040337C
00409995 8B85B0FEFFFF                   mov     eax,[ebp-00000150h]
0040999B E89D9BFFFF                     call    SUB_L00403530
004099A0 50                             push    eax
004099A1 E87AAAFFFF                     call    jmp_KERNEL32.DLL!SetFileAttributesA
004099A6                        L004099A6:
004099A6 8D85A4FEFFFF                   lea     eax,[ebp-0000015Ch]
004099AC 8D562C                         lea     edx,[esi+2Ch]
004099AF B904010000                     mov     ecx,00000104h
004099B4 E85F99FFFF                     call    SUB_L00403318
004099B9 8B8DA4FEFFFF                   mov     ecx,[ebp-0000015Ch]
004099BF 8D85A8FEFFFF                   lea     eax,[ebp-00000158h]
004099C5 8B55FC                         mov     edx,[ebp-04h]
004099C8 E8AF99FFFF                     call    SUB_L0040337C
004099CD 8B85A8FEFFFF                   mov     eax,[ebp-00000158h]
004099D3 E838FBFFFF                     call    SUB_L00409510
004099D8 56                             push    esi
004099D9 57                             push    edi
004099DA E839A9FFFF                     call    jmp_KERNEL32.DLL!FindNextFileA
```

Figure 4: Function to infect executable files

The virus avoids infecting files if any strings in their paths or filenames match a list of hardcoded strings present in the malware (Figure 5).



```
00409F4C                        SLPD0409F4C_system:
00409F4C 73797374656D                   db      'system'
00409F52 0000                           Align   4
00409F54 FFFFFFFF                       dd      FFFFFFFFh
00409F58 08000000                       dd      00000008h
00409F5C                        SLPD0409F5C_system32:
00409F5C 7379737465D3332                db      'system32'
00409F64 00000000                       Align   4
00409F68 FFFFFFFF                       dd      FFFFFFFFh
00409F6C 07000000                       dd      00000007h
00409F70                        SLPD0409F70_windows:
00409F70 77696E646F7773                 db      'windows'
00409F77 00                             Align   4
00409F78 FFFFFFFF                       dd      FFFFFFFFh
00409F7C 16000000                       dd      00000016h
00409F80                        SLPD0409F80_Documents_and_Settings:
00409F80 446F63756D656E747320+          db      'Documents and Settings'
00409F96 0000                           Align   4
00409F98 FFFFFFFF                       dd      FFFFFFFFh
00409F9C 19000000                       dd      00000019h
00409FA0                        SLPD0409FA0_System_Volume_Information:
00409FA0 53797374656D20566F6C+          db      'System Volume Information'
00409FB9 000000                         Align   4
00409FBC FFFFFFFF                       dd      FFFFFFFFh
00409FC0 08000000                       dd      00000008h
00409FC4                        SLPD0409FC4_Recycled:
00409FC4 52656379636C6564               db      'Recycled'
00409FCC 00000000                       Align   4
00409FD0 FFFFFFFF                       dd      FFFFFFFFh
00409FD4 05000000                       dd      00000005h
00409FD8                        SLPD0409FD8_winnt:
00409FD8 77696E6E74                     db      'winnt'
00409FDD 000000                         Align   4
00409FE0 FFFFFFFF                       dd      FFFFFFFFh
00409FE4 0F000000                       dd      0000000Fh
00409FE8                        SLPD0409FE8__Program_Files_:
00409FE8 5C50726F6772616D2046+          db      '\Program Files\'
00409FF7 00                             Align   4
00409FF8 FFFFFFFF                       dd      FFFFFFFFh
00409FFC 0A000000                       dd      0000000Ah
0040A000                        SLPD040A000_Windows_NT:
0040A000 57696E646F7773204E54           db      'Windows NT'
0040A00A 0000                           Align   4
0040A00C FFFFFFFF                       dd      FFFFFFFFh
0040A010 0D000000                       dd      0000000Dh
0040A014                        SLPD040A014_WindowsUpdate:
0040A014 57696E646F7773557064+          db      'WindowsUpdate'
0040A021 000000                         Align   4
0040A024 FFFFFFFF                       dd      FFFFFFFFh
0040A028 14000000                       dd      00000014h
0040A02C                        SLPD040A02C_Windows_Media_Player:
0040A02C 57696E646F7773204D65+          db      'Windows Media Player'
0040A040 00000000                       Align   8
0040A044 FFFFFFFF                       dd      FFFFFFFFh
0040A048 0F000000                       dd      0000000Fh
0040A04C                        SLPD040A04C_Outlook_Express:
0040A04C 4F75746C6F6F6B204578+          db      'Outlook Express'
0040A05B 00                             Align   4
0040A05C FFFFFFFF                       dd      FFFFFFFFh
0040A060 11000000                       dd      00000011h
0040A064                        SLPD040A064_Internet_Explorer:
0040A064 496E7465726E65742045+          db      'Internet Explorer'
0040A075 000000                         Align   4
0040A078 FFFFFFFF                       dd      FFFFFFFFh
0040A07C 14000000                       dd      00000014h
0040A080                        SLPD040A080_ComPlus_Applications:
0040A080 436F6D506C757320417+           db      'ComPlus Applications'
0040A094 00000000                       Align   4
0040A098 FFFFFFFF                       dd      FFFFFFFFh
0040A09C 0A000000                       dd      0000000Ah
0040A0A0                        SLPD040A0A0_NetMeeting:
0040A0A0 4E65744D656574696E67           db      'NetMeeting'
0040A0AA 0000                           Align   4
0040A0AC FFFFFFFF                       dd      FFFFFFFFh
0040A0B0 0C000000                       dd      0000000Ch
0040A0B4                        SLPD040A0B4_Common_Files:
0040A0B4 436F6D6D6F6E2046696C+          db      'Common Files'
0040A0CD 00000000                       Align   8
0040A0C4 FFFFFFFF                       dd      FFFFFFFFh
0040A0C8 09000000                       dd      00000009h
0040A0CC                        SLPD040A0CC_Messenger:
0040A0CC 4D657373656E676572             db      'Messenger'
0040A0D5 000000                         Align   4
0040A0D8 FFFFFFFF                       dd      FFFFFFFFh
0040A0DC 10000000                       dd      00000010h
0040A0E0                        SLPD040A0E0_Microsoft_Office:
0040A0E0 4D6963726F736F667420+          db      'Microsoft Office'
0040A0F0 00000000                       Align   8
0040A0F4 FFFFFFFF                       dd      FFFFFFFFh
0040A0F8 26000000                       dd      00000026h
```

Figure 5: Strings avoided in File & path names during infection

- **Peristence**

The malware adds the following registry value to achieve automatic execution at every system startup:

```
HKU\S-1-5-21-1469219667-2966600018-1877350481-1001\Software\Microsoft\Windows NT\CurrentVersion\Windows\load: "C:\Windows\rundl132.exe"
```

- **Antivirus Termination**

It also terminates 'Kingsoft AntiVirus Service' to evade detection using the command line (Figure 6). The following command is executed:
'net stop "Kingsoft AntiVirus Service"'

```
00407A8A                    L00407A8A:
00407A8A E8F5260000         call        SUB_L0040A184
00407A8F 84C0               test        al,al
00407A91 0F85E2000000       jnz         L00407B79
00407A97 E83CECFFFF         call        SUB_L004066D8
00407A9C E8B3E7FFFF         call        SUB_L00406254
00407AA1 6A00               push        00000000h
00407AA3 68BC7B4000         push        SSZ00407BBC_net_stop__Kingsoft_AntiVirus_Ser
00407AA8 E8CBC9FFFF         call        jmp_KERNEL32.DLL!WinExec
00407AAD B8EC7B4000         mov         eax,SLP00407BEC_6_
00407AB2 E8D9DDFFFF         call        SUB_L00404B90
00407AB7 C6052D9041000D     mov         byte ptr [L00419020],00h
00407ABE 8D45FC             lea         eax,[ebp-04h]
00407AC1 50                 push        eax
00407AC2 6A00               push        00000000h
00407AC4 6A00               push        00000000h
00407AC6 6864A34000         push        L0040A364
00407ACB 6A00               push        00000000h
00407ACD 6A00               push        00000000h
00407ACF E804C8FFFF         call        jmp_KERNEL32.DLL!CreateThread
00407AD4 A160B14000         mov         eax,[L0040B160]
00407AD9 F6401C01           test        byte ptr [eax+1Ch],01h
00407ADD 741F               jz          L00407AFE
00407ADF C6052190410000     mov         byte ptr [L00419021],00h
00407AE6 8D45F8             lea         eax,[ebp-08h]
00407AE9 50                 push        eax
00407AEA 6A00               push        00000000h
00407AEC 6A00               push        00000000h
00407AEE 68A8A44000         push        L0040A4A8
00407AF3 6A00               push        00000000h
00407AF5 6A00               push        00000000h
00407AF7 E8DCC7FFFF         call        jmp_KERNEL32.DLL!CreateThread
00407AFC EB07               jmp         L00407B05
00407AFE                    L00407AFE:
00407AFE C6052190410001     mov         byte ptr [L00419021],01h
```

Figure 6: Antivirus Termination

- **Network Communication**

The virus announces the infection to a controlled server, '154.92.223.77' by sending a 'Hello World' packet. The network communication is established by communicating over ICMP packets (Figure 7).
Viking also tries to propagate via network shares & copies itself to shared folders, 'admin$' & 'ipc$' (Figure 8) with administrator & guest accounts.

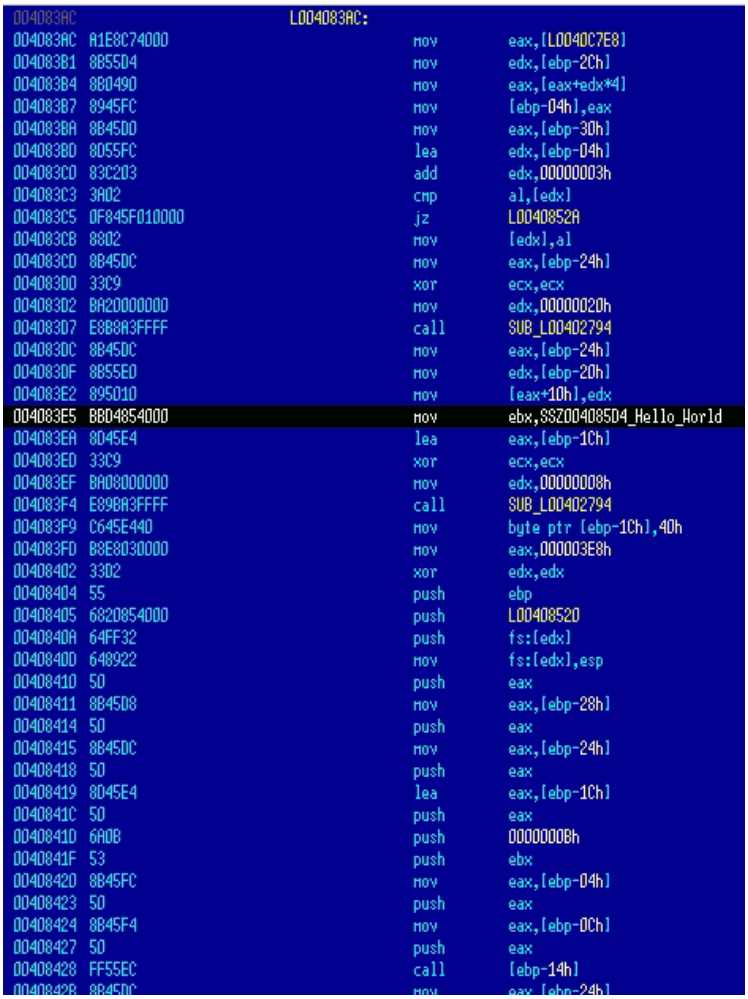


Figure 7: 'Hello_World' packet sent to announce infection

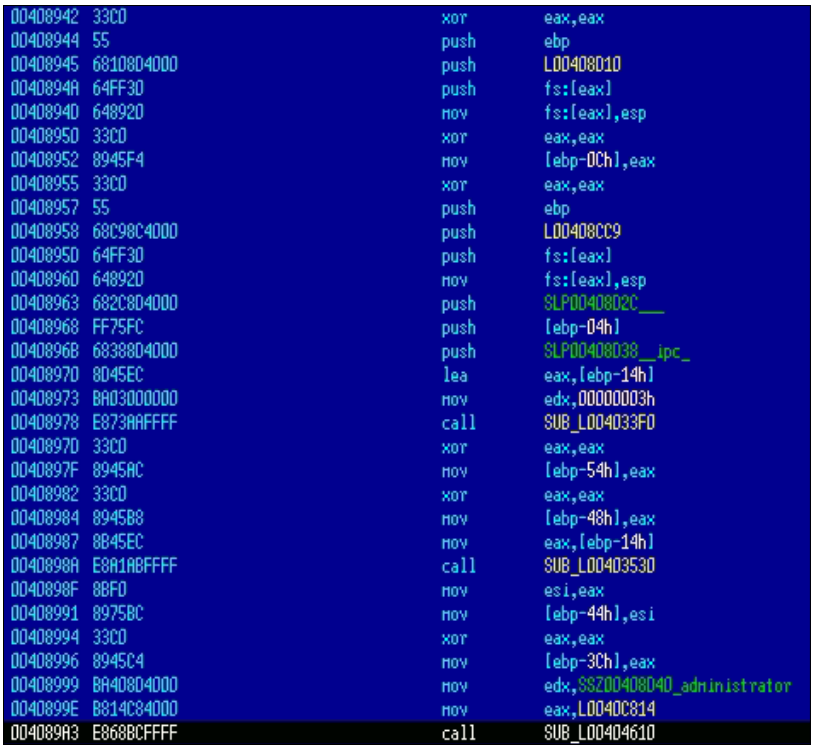


Figure 8: Viking copies itself to shared folders for network propagation

- **Keylogger Capabilities**

The DLL component of Viking runs a keylogger (Figure 9) & captures user input which is later exfiltrated to the C2 server, along with the victim system information including the computer name & processor specifications.

```
;
00402885  8D4000                          Align     4
00402888                          jmp_USER32.DLL!GetKeyboardType:
00402888  FF2540A14100                    jmp       [USER32.DLL!GetKeyboardType]
0040288E  8BC0                            Align     4
00402890                          SUB_L00402890:
00402890  53                              push      ebx
00402891  33DB                            xor       ebx,ebx
00402893  6A00                            push      00000000h
00402895  E8EEFFFFFF                      call      jmp_USER32.DLL!GetKeyboardType
0040289A  83F807                          cmp       eax,00000007h
0040289D  751C                            jnz       L004028BB
0040289F  6A01                            push      00000001h
004028A1  E8E2FFFFFF                      call      jmp_USER32.DLL!GetKeyboardType
004028A6  2500FF0000                      and       eax,0000FF00h
004028AB  3D00000000                      cmp       eax,00000000h
004028B0  7407                            jz        L004028B9
004028B2  3D00040000                      cmp       eax,00000400h
004028B7  7502                            jnz       L004028BB
004028B9                          L004028B9:
004028B9  B301                            mov       bl,01h
004028BB                          L004028BB:
004028BB  8BC3                            mov       eax,ebx
004028BD  5B                              pop       ebx
004028BE  C3                              retn
```

Figure 9: DLL component runs a Keylogger

Subex Secure Protection

Subex Secure detects the Viking prepending virus as 'SS_Gen_Viking_PE_A'.

Sample Details

1.	2694e399ed931a811067f05499aed0e14184d369ba50476262d8f3d42ff43fe4
2.	ace170a3928a41b38026017f5cd4ec42d3bfd900f21f44c2a1e6d53d13edad80
3.	f1fe3a98cf3017a037ede5104baefddfe120513c22d3cc6c398fa50732103ec1
4.	dd3dbe3b41afffedfb52705278c095755a99031900f2397eeb509eb30f9e0301

Analysed Sample

Viking Main Executable	d9eade75e518762affc09f318c944fef
Embedded Exe (Logo1_.exe)	5ad9d686420dc3fa331af7603a5fd3de
Embedded Exe (rundl132.exe)	5b1123cfbb6e5ef974a6d484766bb6fe
DLL Component (vdll.dll)	44d0d7cb8233379ae1a0e2190faf720d

Network Communication

1.	154.92.223.77

MITRE Attack Techniques

TACTIC	ID	NAME
Execution	T1173	Dynamic Data Exchange
Execution	T1059	Command Line Interface
Privilege Escalation	T1055	Process Injection

Persistence	T1179	Hooking
Persistence	T1215	Kernel Module & Extensions
Defence Evasion	T1112	Modify Registry
Defence Evasion	T1036`	Masquerading
Discovery	T1057	Process Discovery
Discovery	T1083	File and Directory Discovery
Discovery	T1010	Application Window Discovery
Discovery	T1012	Query Registry
Privilege Escalation	T1076	Remote Desktop Protocol
Execution	T1043	Commonly Used Port
Discovery	T1489	Service Stop
Discovery	T1082	System Information Discovery
Collection	T1056.001	Keylogger

Our Honeypot Network

This report has been prepared from threat intelligence gathered by our honeypot network. This honeypot network is today operational in 62 cities across the world. These cities have at least one of these attributes:

- Are landing Centers for submarine cables
- Are internet traffic hotspots
- House multiple IoT projects with a high number of connected endpoints
- House multiple connected critical infrastructure projects
- Have academic and research Centers focusing on IoT
- Have the potential to host multiple IoT projects across domains in the future

Over 3.5 million attacks a day is being registered across this network of individual honeypots. These attacks are studied, analysed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Subex. The honeypot network includes over 4000 physical and virtual devices covering over 400 device architectures and varied connectivity mediums globally. These devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.