CosmicDuke Malware Analysis - CYFIRMA

Archived: 2026-04-05 22:40:36 UTC

Published On : 2022-08-29

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.

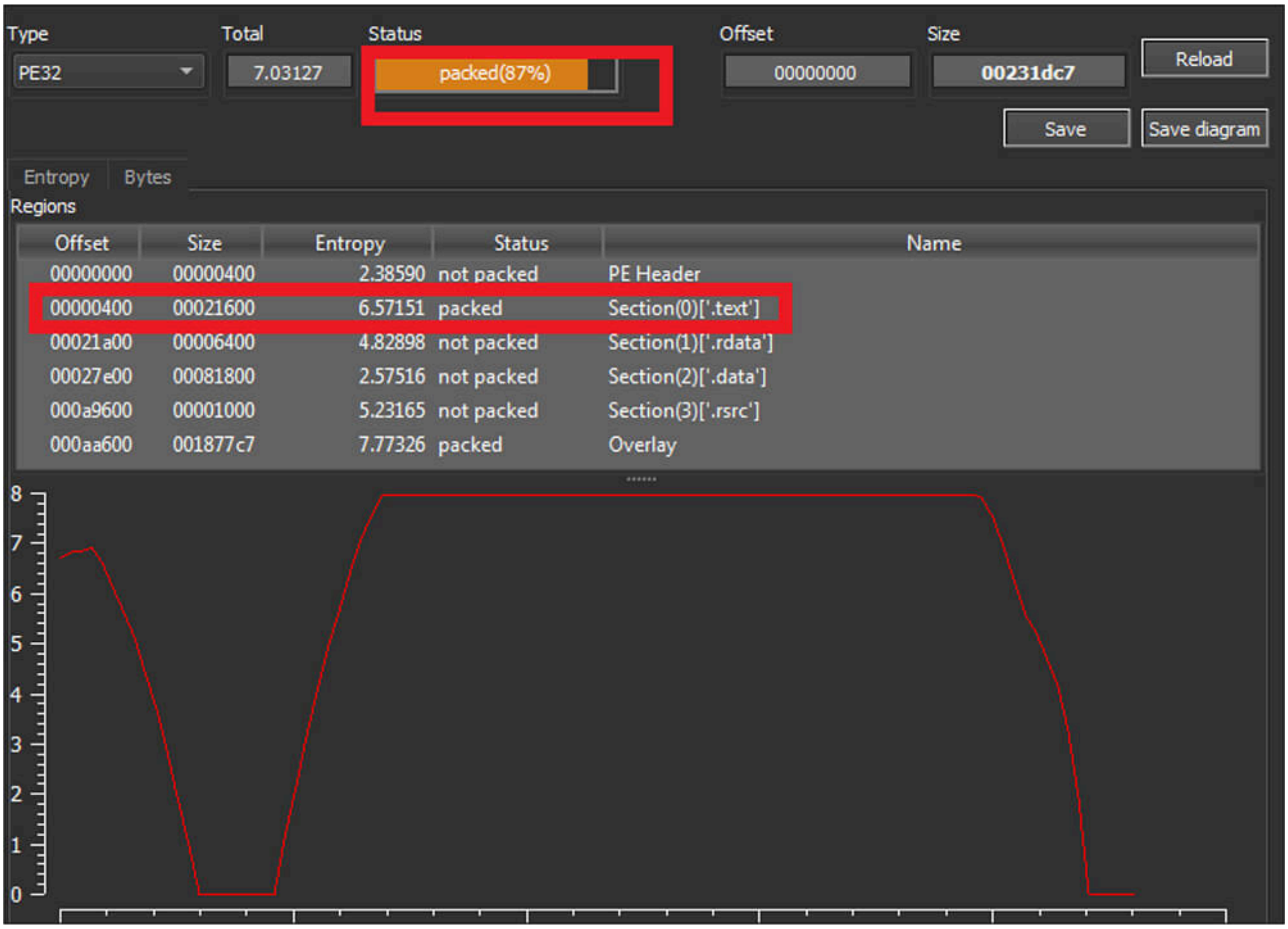
One of the campaigns 'natural disaster' which is potentially active since 17 March 2022 with the motive of exfiltration of sensitive databases, and customer information for financial gains. The threat actor is suspected to leverage attack methods such as exploiting the weakness in the systems, phishing 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.



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”

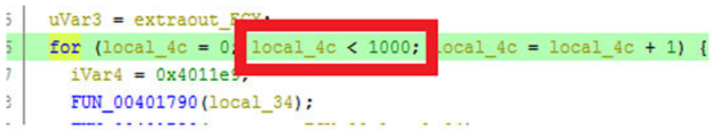
Network Communication

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:

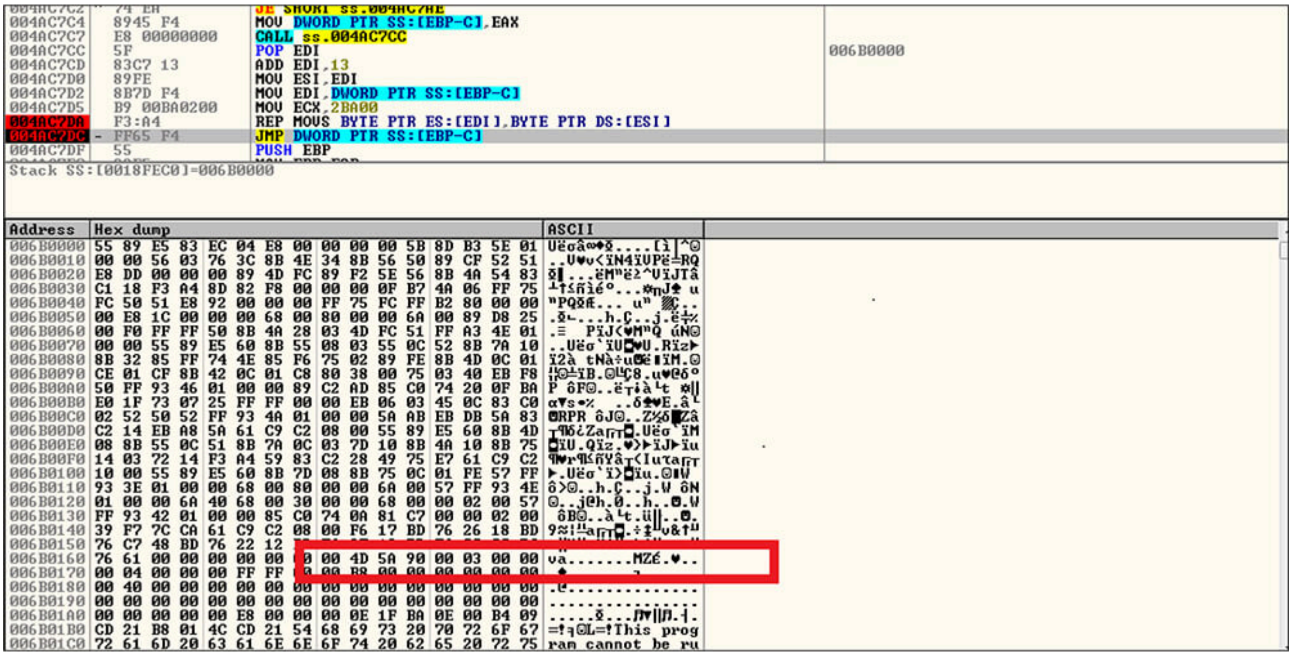
- 199[.]231[.]188[.]109
- 46[.]246[.]120[.]178

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.



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.



1st Stage Payload (unpacked)

Sample Details:

MD5: 335D2EE728B4C1591B5B374A7CE4B758

SHA256: 42AFD884116DF2267696DA88827E8F774155C8B1DA86BCE968BE20765EB8BB7C File Type: Windows PE

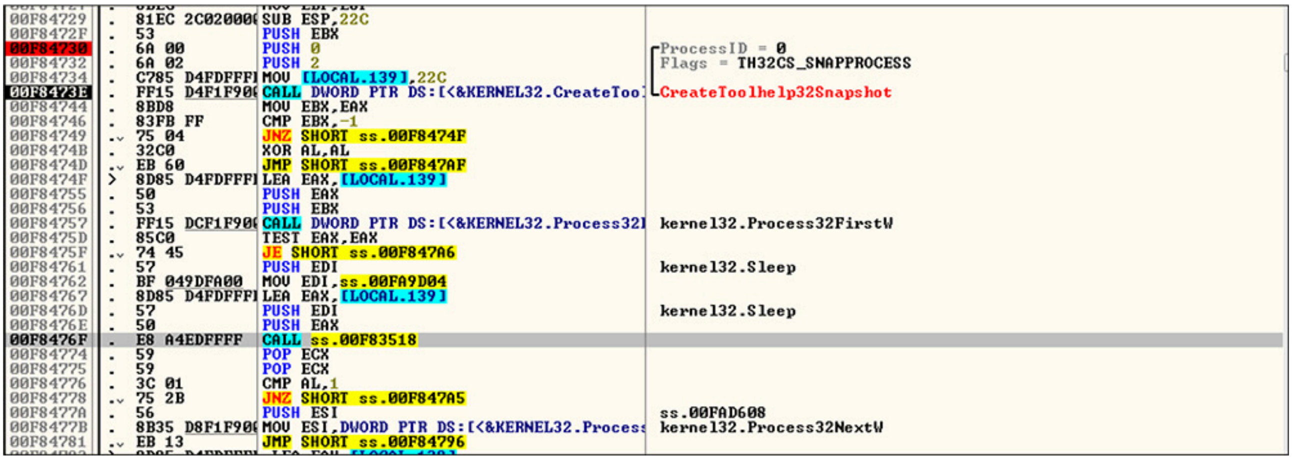
Architecture: 32 Bit

Subsystem: GUI

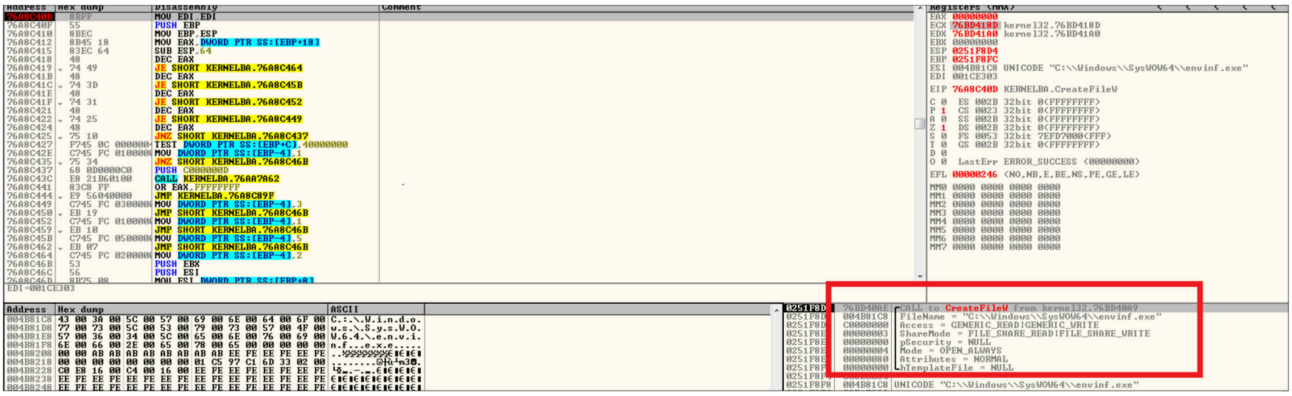
This malware sample was also written in Microsoft Visual C++ programming language. This malware binary file's size is 294551 (bytes). As shown below, this file is having the version information as Microsoft Corporation [internal file name is svchost.exe], with this trick allowing the threat actor to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

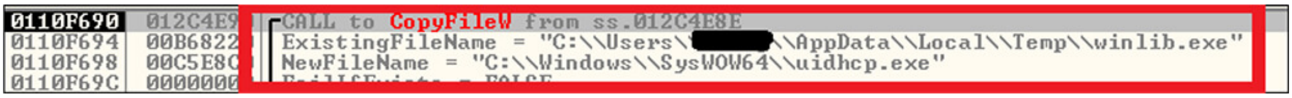
This CosmicDuke backdoor loader initially verifies any security product running in the victim system before executing the CosmicDuke malware activity by calling CreateToolhelp32Snapshot, Process32Next, and Process32First. If any security product is running, this malware will be terminated with no expression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



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.

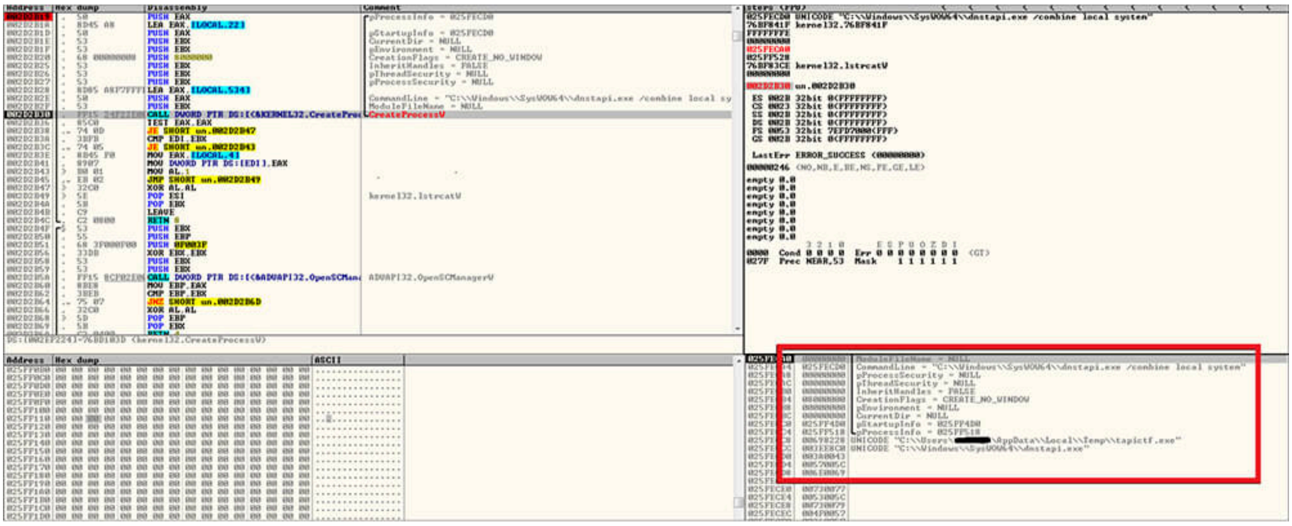


Once the unpacked file is created in the system32, this malicious binary obtains the temp folder location by calling GetTempPathW, then creates a 5kb file [File hash: 0499C600266D8311722BBC31B89FB9AC] by calling again CreateFileW, after that this 5kb file is copied into the system folder by calling CopyFileW.

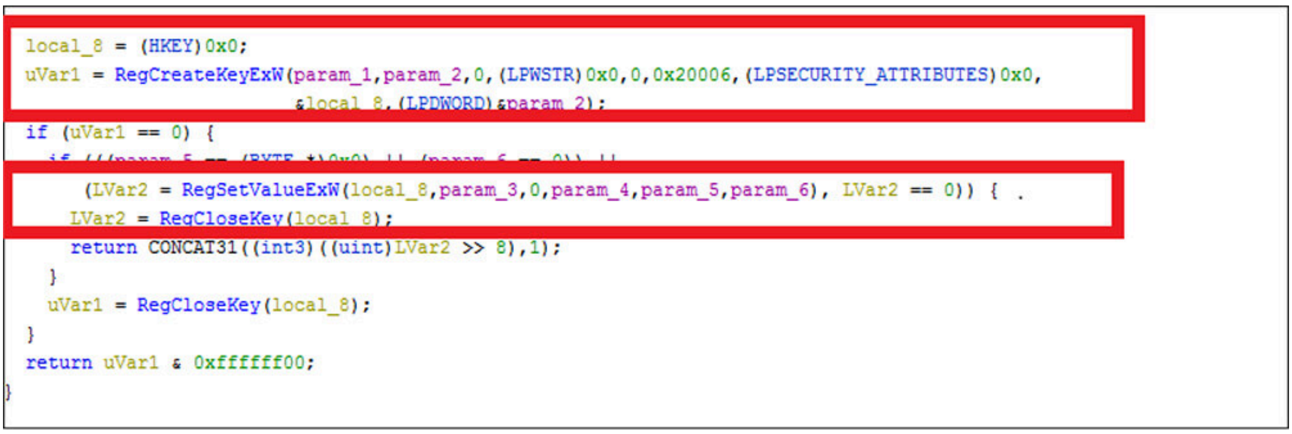


Similar to the above behavior, this malware code creates a 4kb file in the temp folder [file hash: 943E98CB74058DFA942D9D6184E936B1] after that copies this file to system32 as .scr file extension.

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



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.



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

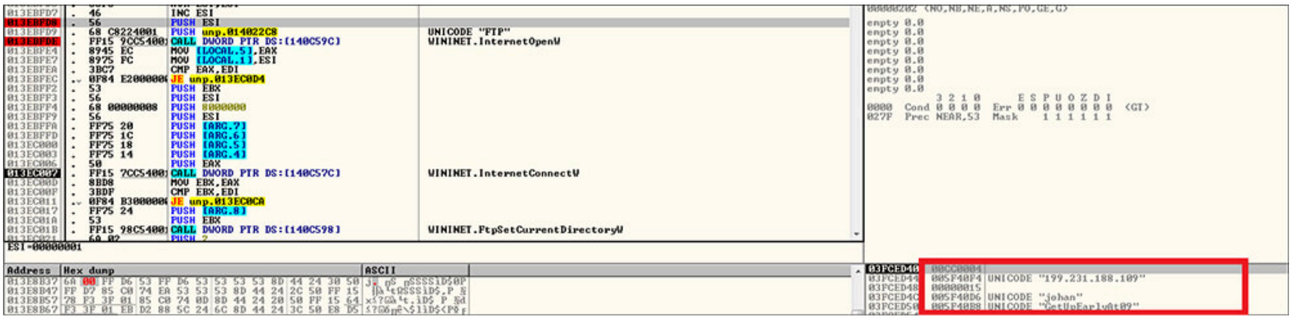
```
012CA1A1  .   56            PUSH ESI                                   ss.012EAEE0
012CA1A2  .   33ED          XOR EBP,EBP
012CA1A4  .   57            PUSH EDI
012CA1A5  .   33F6          XOR ESI,ESI                                ss.012EAEE0
012CA1A7  >   FF15 90F32D01 CALL DWORD PTR DS:[<&USER32.GetForegrour  [GetForegroundWindow
012CA1AD  .   50            PUSH EAX                                   [hWnd = 00000001
012CA1AE  .   FF15 84F32D01 CALL DWORD PTR DS:[<&USER32.OpenClipboai  [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.GetClipboarc  [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();
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.



Dropped file_01

Sample Details:

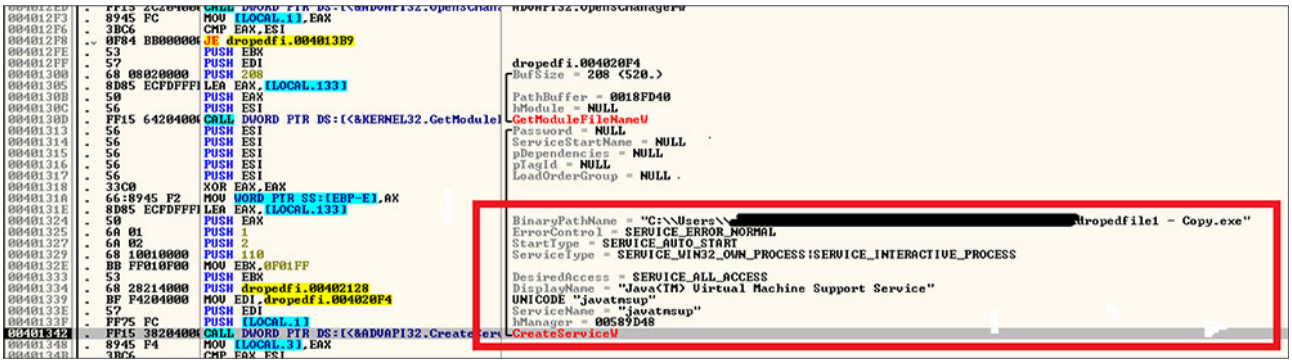
MD5: 0499C600266D8311722BBC31B89FB9AC

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].



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