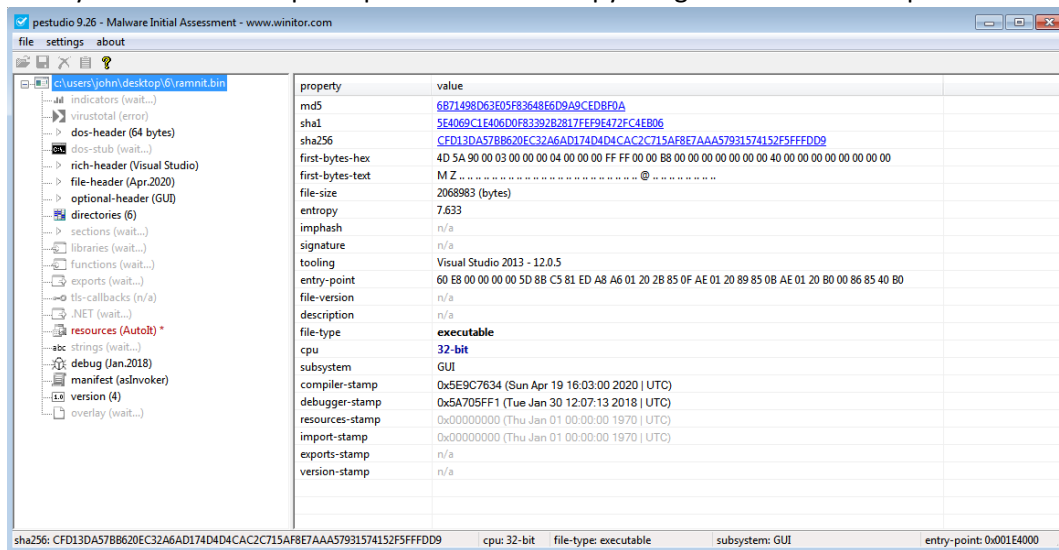
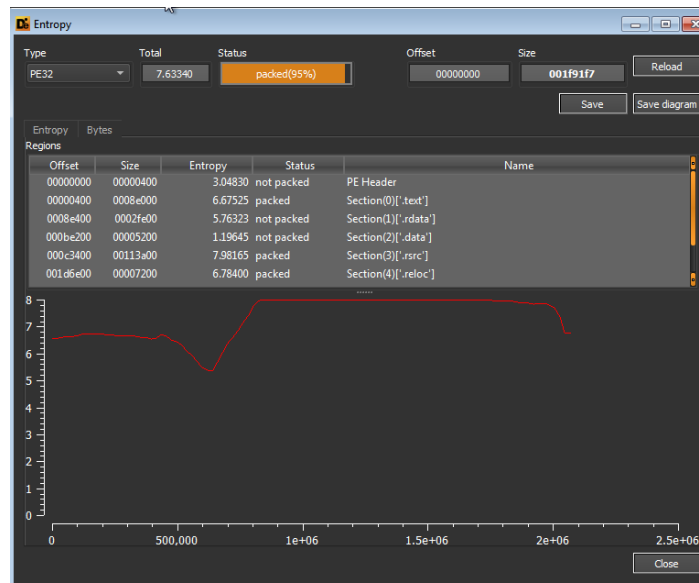


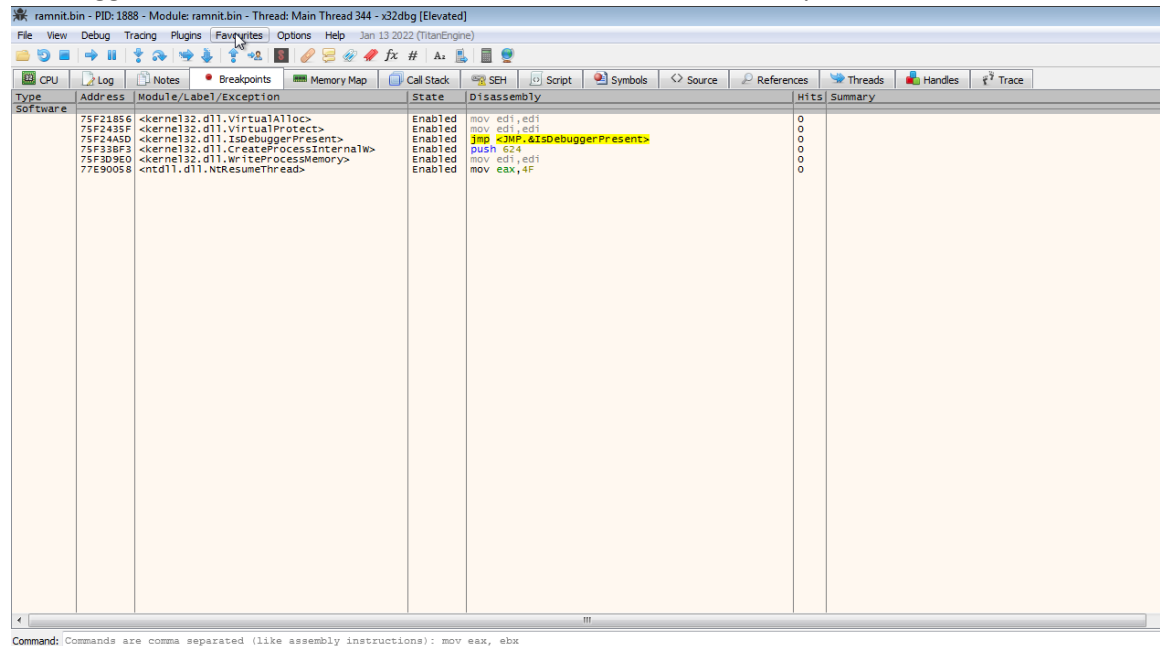
Initially we load the sample in pestudio and entropy is high which show it is packed.



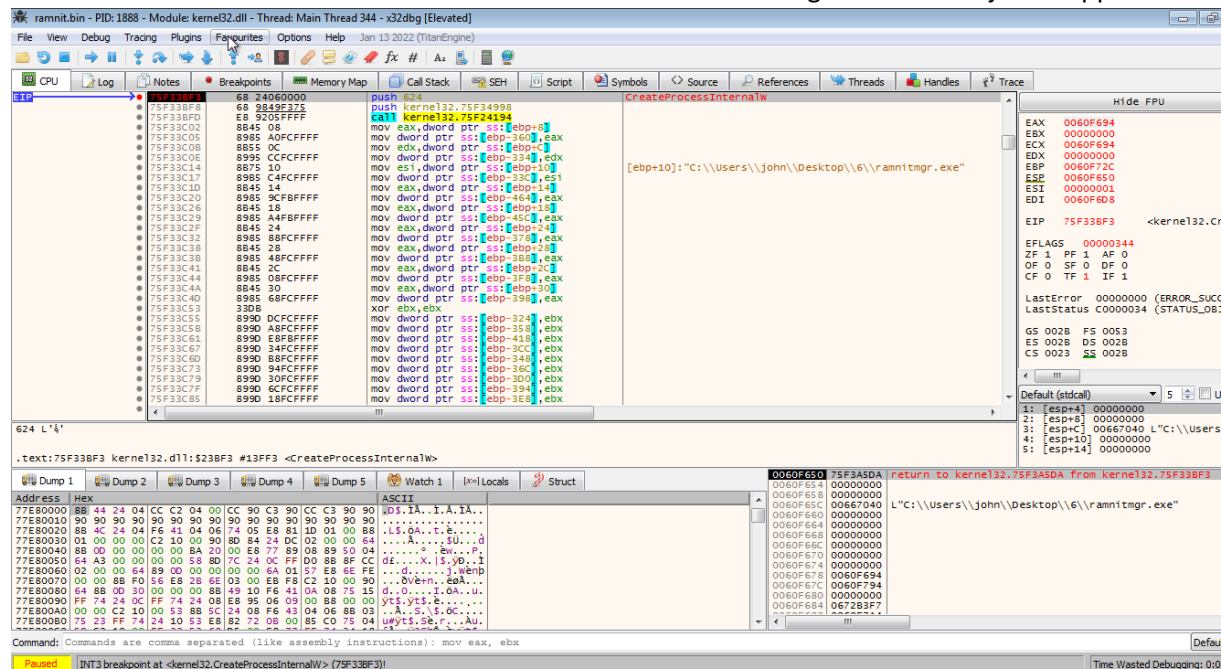
Now we loaded the sample in DetectItEasy and view the entropy i.e.7.63 and it is showing the sample status as packed.



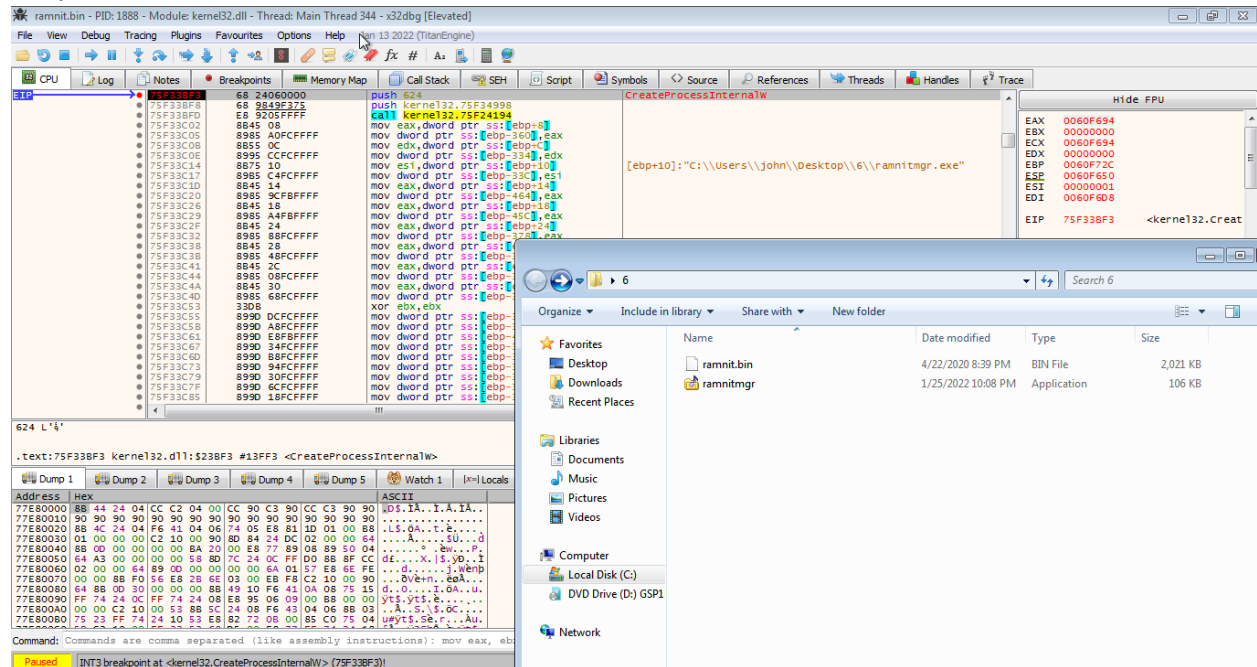
Now we will load the sample in x32dbg and put breakpoint on VirtualAlloc, VirtualProtect, IsDebuggerPresent, CreateProcessInternalW, WriteProcessMemory, NtResumeThread.



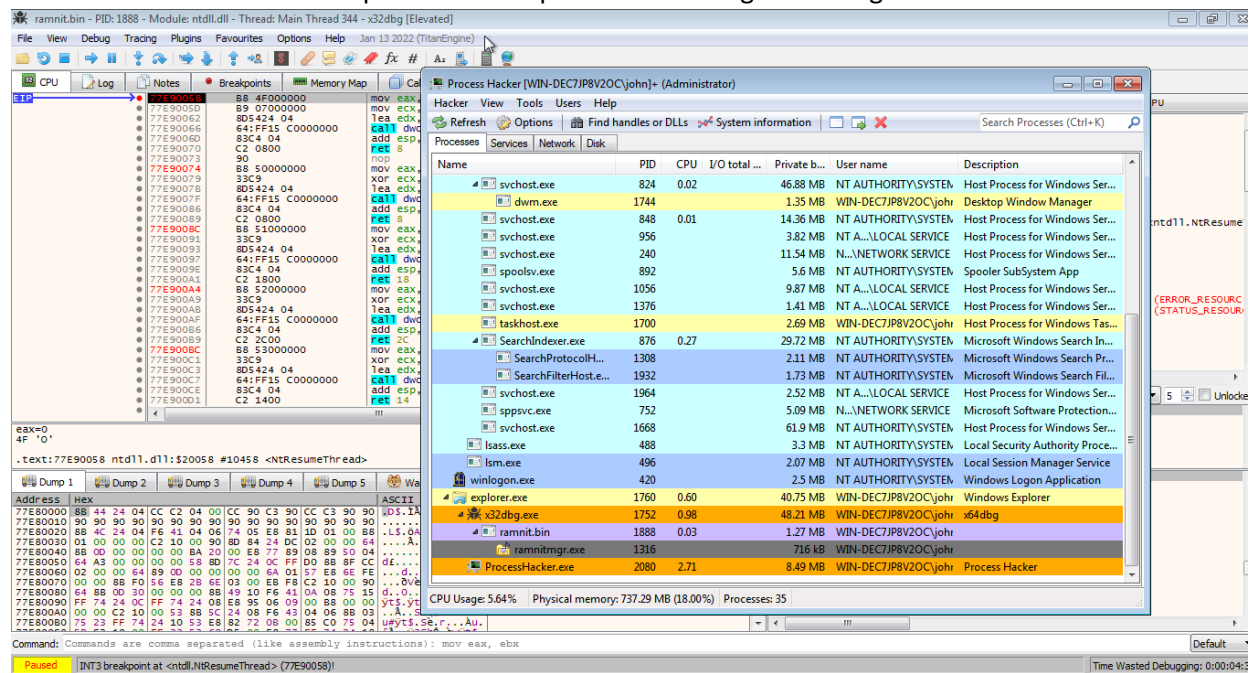
Now we hit the CreateProcessInternalW it will run the file ramnitgr.exe which is just dropped.



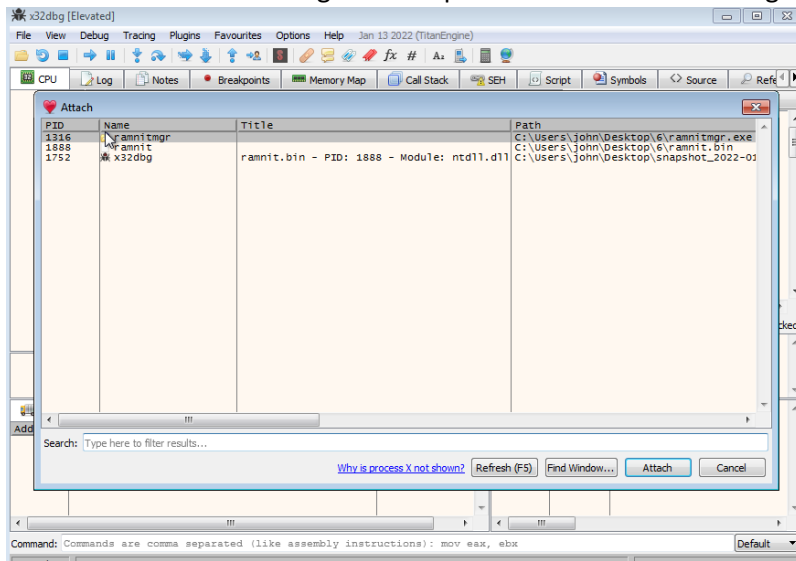
Here is the dropped file, Now to run this API we will click on “run to user” code, we can verify it using the process hacker.



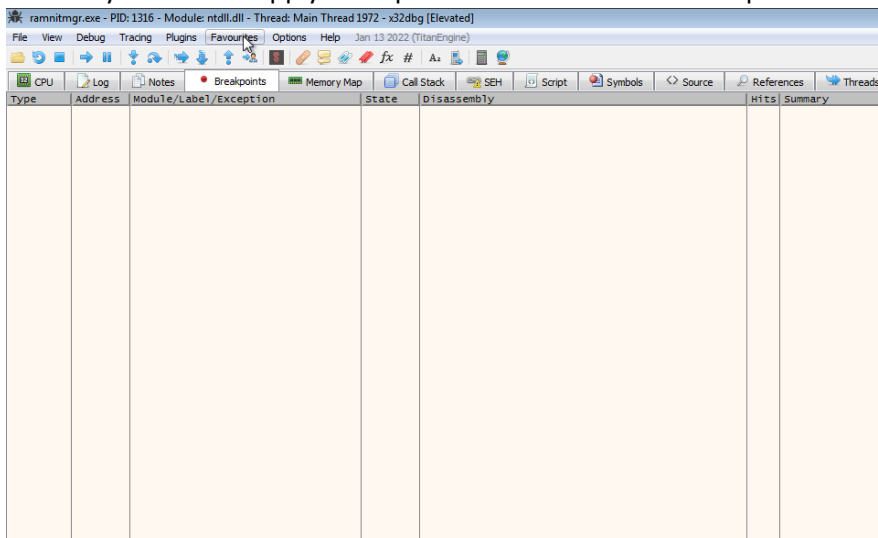
Here we can see ramnit had spawn the new process ramnitmgr.exe using API `CreateProcessInternalW`.



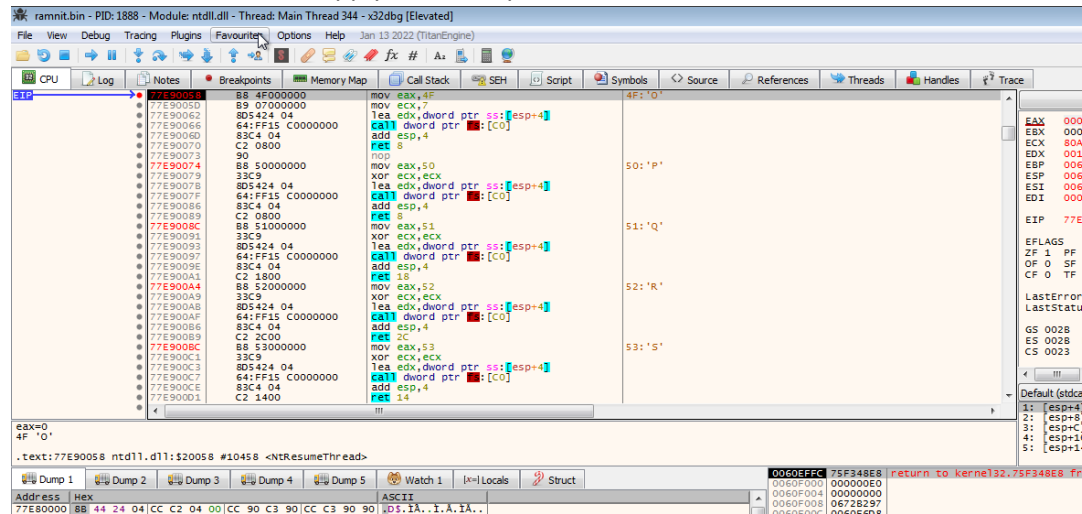
Since we know it is running we will open new instance of x32dbg and attach it.



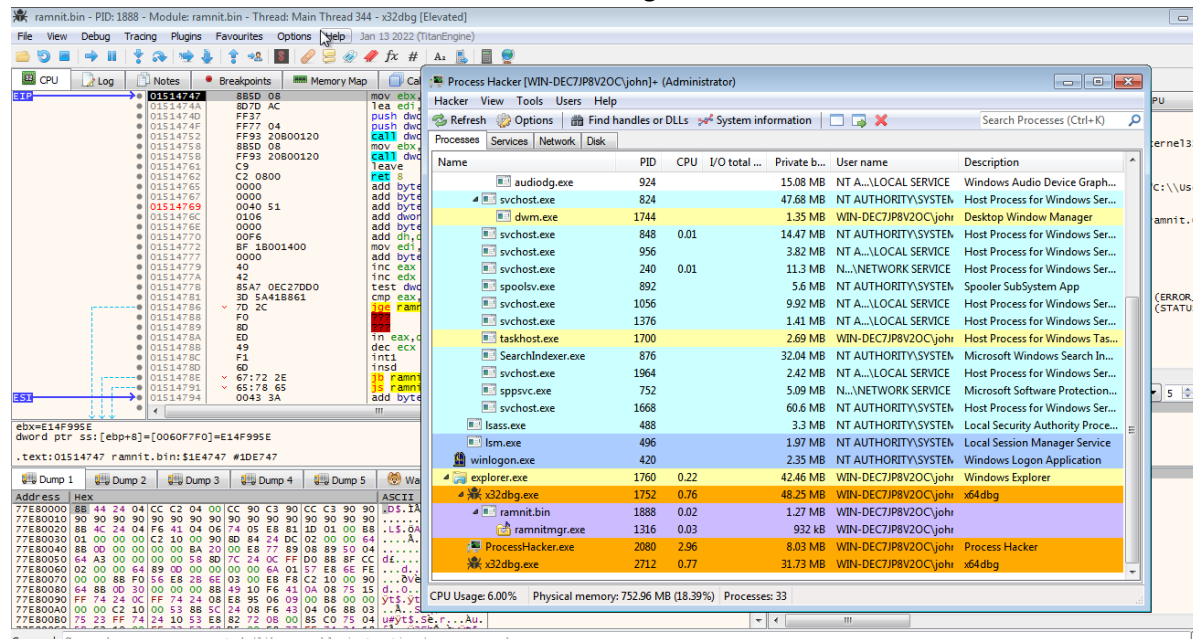
Currently we cannot apply breakpoints here because the new process is currently in suspended state.



Now here switch to the initially loaded sample and clicked the “run to user code” this make the API to run and after that we can apply the breakpoints.



Here we can see in Process Hacker it is now in running state.



Here we have applied all the breakpoints press f9.

ramnitmgr.exe - PID: 1316 - Module: ntdll.dll - Thread: Main Thread 1972 - x32dbg [Elevated]						
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)						
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads						
Type	Address	Module/Label/Exception	State	Disassembly	Hits	Summary
Software	75F21896	<kernel32.dll.VirtualAlloc>	Enabled	mov edi,edi	0	
	75F2439F	<kernel32.dll.VirtualProtect>	Enabled	mov edi,edi	0	
	75F2445D	<kernel32.dll.IsDebuggerPresent>	Enabled	jmp <JMP.&IsDebuggerPresent>	0	
	75F338F3	<kernel32.dll.CreateProcessInternal>	Enabled	push 624	0	
	75F3D9E0	<kernel32.dll.WriteProcessMemory>	Enabled	mov edi,edi	0	

Now we hit the IsDebuggerPresent press the “execute till return” and modify the value returned in eax to 0.

ramnitmgr.exe - PID: 1888 - Module: kernel32.dll - Thread: Main Thread 344 - x32dbg [Elevated]						
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)						
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace						
JMP	75F2445D	EB 05	jmp <JMP.&IsDebuggerPresent>	IsDebuggerPresent		
	75F2445F	90	nop			
	75F24460	90	nop			
	75F24461	90	nop			
	75F24462	90	nop			
	75F24463	90	nop			
	75F24464	FF25 940DF275	jmp dword ptr ds:[&IsDebuggerPresent]	JMP.&IsDebuggerPresent		
	75F2446B	90	nop			
	75F2446C	90	nop			
	75F2446D	90	nop			
	75F2446E	90	nop			
	75F2446F	8BFF	mov edi,edi			
	75F24471	55	push ebp			
	75F24472	8BEC	mov ebp,esp			
	75F24474	5D	pop ebp			
	75F24475	EB 05	jmp <JMP.&GetModuleHandleExW>	GetModuleHandleExW		
	75F24477	90	nop			
	75F24478	90	nop			
	75F24479	90	nop			
	75F2447A	90	nop			
	75F2447B	90	nop			
	75F2447C	FF25 180BF275	jmp dword ptr ds:[&GetModuleHandleExW]	JMP.&GetModuleHandleExW		
	75F24482	90	nop			
	75F24483	90	nop			
	75F24484	90	nop			
	75F24485	90	nop			
	75F24486	90	nop			
	75F24487	6A 48	push 48			
	75F24489	68 E048F275	push kernel32.75F248E0			
	75F2448E	E8 5DCBF2FF	call kernel32.75F215F0			
<JMP.&IsDebuggerPresent>						
.text:75F2445D kernel32.dll:\$14A5D #4E5D <IsDebuggerPresent>						
Dump 1 Hex Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct						
Address	Hex	ASCII				
77E80000	8B 44 24 04 CC C2 04 00 CC 90 C3 90 CC C3 90 90	ds.IA..I.A.IA..				
77E80010	90 90 90 90 90 90 90 90 90 90 90 90 90 90				
77E80020	8B 4C 24 04 F6 41 04 06 74 05 E8 81 10 01 00 88	..S.A..t.e....				
77E80030	01 00 00 00 C2 10 00 00 80 84 24 DC 02 00 00 64A....S...d				
77E80040	88 00 00 00 00 00 8A 20 00 E8 77 89 08 89 50 04P.....				
77E80050	64 A3 00 00 00 00 58 80 7C 24 0C FF D0 8B 8F CC	d...X..s.yd..I				
77E80060	02 00 00 64 89 00 00 00 00 00 6A 01 57 E8 8E FE	...d....j.wenp				
77E80070	00 8B F6 16 E8 2B 66 03 00 E8 F8 C2 10 00 90	...v.h.n..caA..				
77E80080	64 88 00 30 00 00 00 8B 49 10 F6 41 0A 08 75 15	d..O...I.OA..u.				
77E80090	FF 74 24 0C FF 74 24 08 E8 95 06 09 00 88 00 00	Yt..Yt..e....				
77E800A0	00 00 C2 10 00 53 8C 24 08 F6 43 04 06 8B 03	..A..S..t..OC..				
77E800B0	75 23 FF 74 24 10 53 E8 82 72 0B 00 85 C0 75 04	uwytS..Se.r...Au				
Command: Commands are comma separated (like assembly instructions): mov eax, ebx						

0060F0FC 71DA4F05 return to uxtheme.71DA4F05 from ???

0060F100 00000000

0060F104 00000158

0060F108 0060F128

0060F10C 71DA4598 return to uxtheme.71DA4598 from uxtheme.71DA4598

0060F110 00000000

0060F114 0060F140

0060F118 765E0280 user32.765E0280

0060F11C 00000000

0060F120 00000000

0060F124 00000000

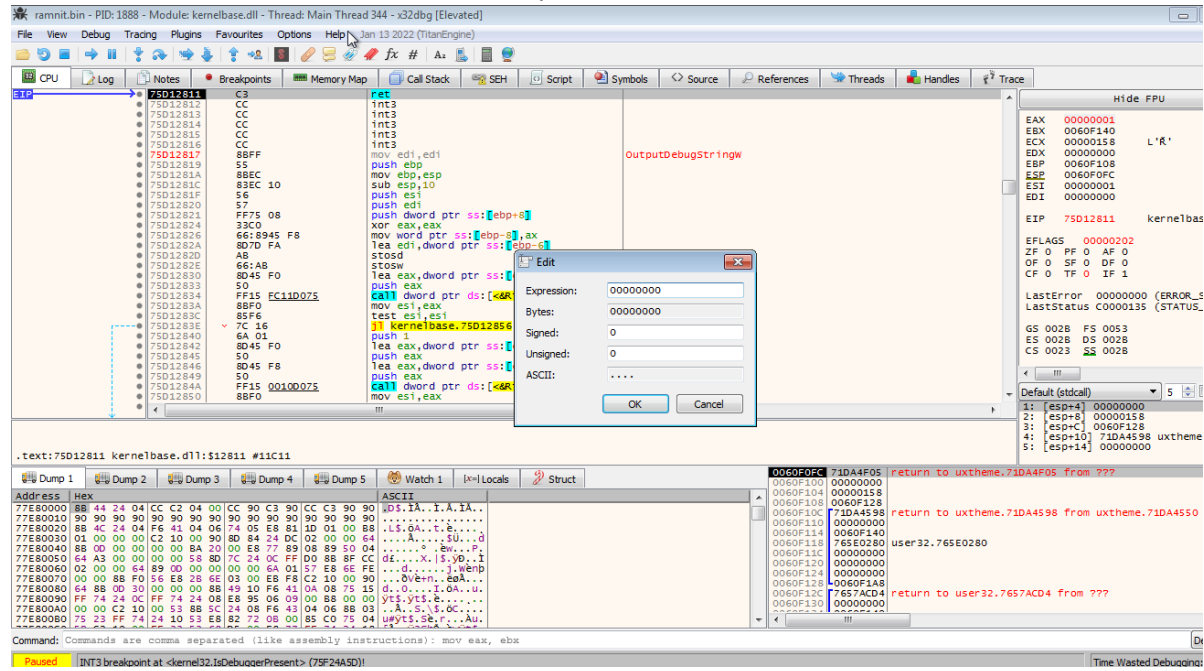
0060F128 0060F1A8

0060F12C 7657ACD4 return to user32.7657ACD4 from ???

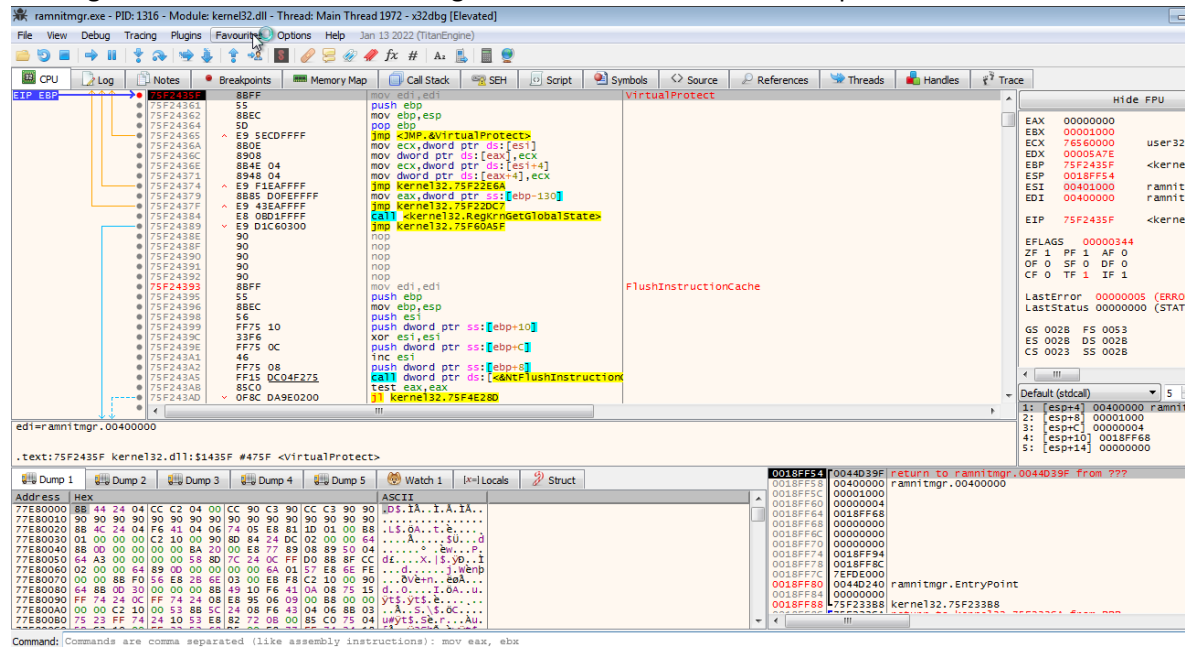
0060F130 00000000

0060F134 00000000

Here we modified the return value to 0 and press f9 it will continue to run.



Now again we switch to the ramnitmgr.exe and here we hit the breakpoint at VirtualProtect.



ramnitrng.exe - PID:1316 - Module:kernelbase.dll Thread: Main Thread 1972 - x32dbg [Elevated]

File View Debug Tracing Plugins Favourites **Registers** Help Jan 13 2022 (TitanEngine)

CPU Log Notes Breakpoints Memory Map Call Stack **DBG** Script Symbols Source References Threads Handles Trace

75D0E326 8BFF mov edi,edi
 75D0E328 55 push ebp
 75D0E329 8BEC mov ebp,esp
 75D0E32B FF75 14 push dword ptr ss:[ebp+14]
 75D0E32E FF75 0C push dword ptr ss:[ebp+0C]
 75D0E331 FF75 08 push dword ptr ss:[ebp+08]
 75D0E334 6AFF push ffffffff
 75D0E339 **CALL kernelbase.VirtualProtectEx**
 75D0E33E C2 1000 pop ebp
 75D0E342 CC int3
 75D0E343 CC int3
 75D0E344 CC int3
 75D0E345 CC int3
 75D0E346 CC int3
 75D0E347 8BFF mov edi,edi
 75D0E349 55 push ebp
 75D0E34B 8BEC mov ebp,esp
 75D0E34D FF75 10 push dword ptr ss:[ebp+10]
 75D0E34F FF75 0C push dword ptr ss:[ebp+0C]
 75D0E352 FF75 08 push dword ptr ss:[ebp+08]
 75D0E355 6AFF push ffffffff
 75D0E357 **CALL kernelbase.VirtualQueryEx**
 75D0E35C 50 pop ebp
 75D0E35D C2 0C00 ret c
 75D0E361 CC int3
 75D0E362 CC int3
 75D0E363 CC int3
 75D0E364 CC int3

kernelbase.VirtualProtectEx

.text:75D0E339 kernelbase.dll:SE339 #D739

Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	Watch 1	Locals	Struct
Address	Hex	Hex			ASCII		
77E80000	8B 44 24 04	CC C2 04 00	CC 90 C3 90	CC C3 90 90	ASCII		
77E80010	90 90 90 90	90 90 90 90	90 90 90 90	90 90 90 90	ASCII		
77E80020	8B 4C 24 04	F6 41 04 06	74 05 E8 81	1D 01 00	ASCII		
77E80030	01 00 00 00	C2 10 04 00	80 C2 10 00	00 00 64	ASCII		
77E80040	8B 00 00 00	00 00 84 2E	00 00 89 08	10 00 00	ASCII		
77E80050	64 A3 00 00	00 00 5B 80	7C 24 00 FF	D0 8B FC	ASCII		
77E80060	02 00 00 00	00 00 00 00	00 00 6A 01	51 E8 FE	ASCII		
77E80070	00 8B F0	2B 2E 03 00	CC 00 EB	00 00 00	ASCII		

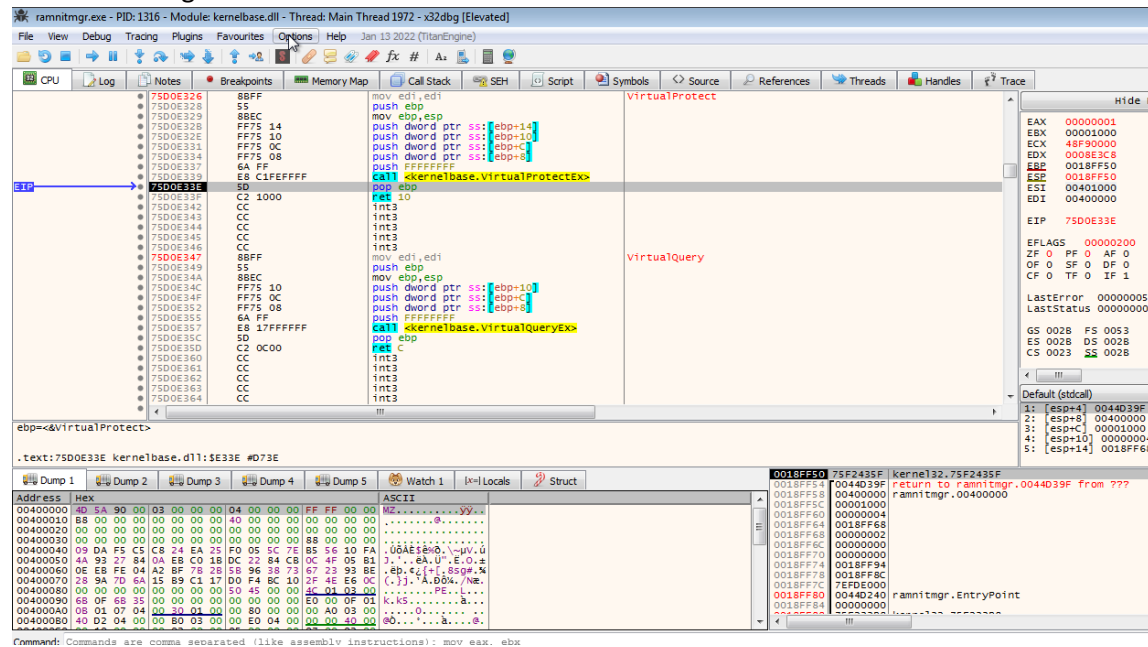
ramtmg.exe - PID: 1316 - Module: kernelbase.dll - Thread: Main Thread 1972 - x32dbg [Elevated]

File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)

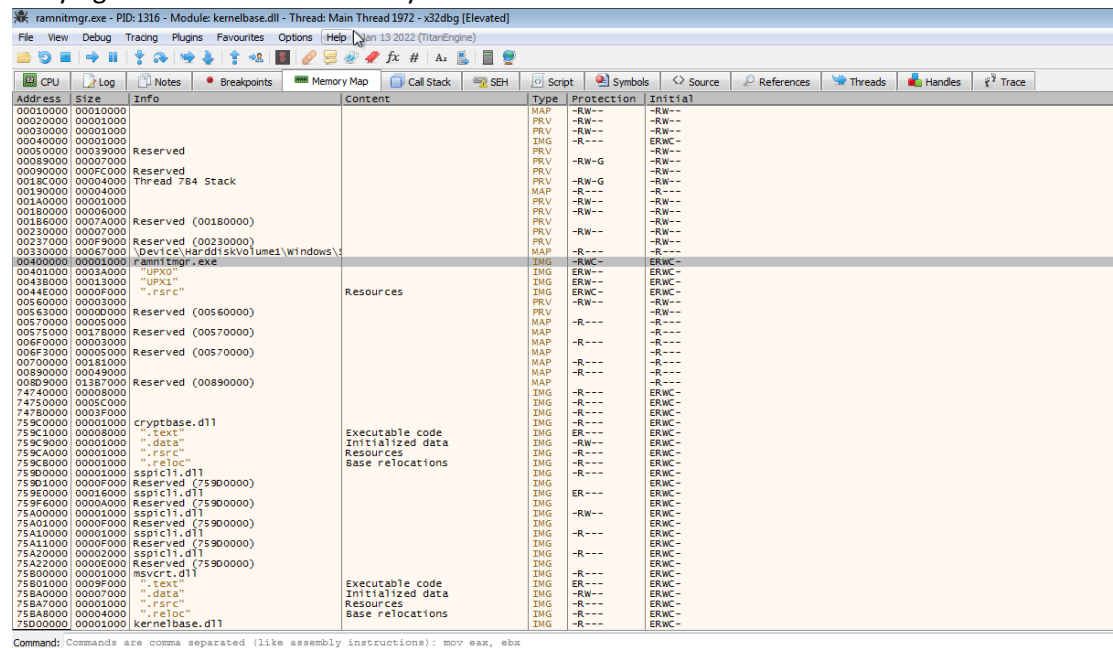
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

Address	Size	Info	Content	Type	Protection	Initial
00010000	00010000			MAP	-RW-	-RW-
00020000	00001000			PRV	-RW-	-RW-
00030000	00001000			PRV	-RW-	-RW-
00040000	00001000			IMG	-R--	ERWC
00050000	00039000	Reserved		PRV	-RW-G	-RW-
00060000	00007000			PRV	-RW-	-RW-
00090000	000FC000	Reserved		PRV	-RW-	-RW-
0018C000	00004000	Thread 784 Stack		PRV	-RW-G	-RW-
00190000	00004000			MAP	-R--	-R--
001A0000	00001000			PRV	-RW-	-RW-
001B0000	00006000			PRV	-RW-	-RW-
001B6000	0007A000	Reserved (001B0000)		PRV	-RW-	-RW-
00230000	00070000			PRV	-RW-	-RW-
00237000	000F9000	Reserved (00230000)		PRV	-RW-	-RW-
00330000	00067000	(Device HarddiskVolume1\Windows\...		MAP	-R--	-R--
00400000	00001000	ramtmg.exe		IMG	-RW-	ERWC
00401000	0003A000	"UPX0"		IMG	ERW-	ERWC
0043B000	00013000	"UPX1"		IMG	ERW-	ERWC
0044E000	0000F000	"rsrc"		IMG	ERWC	ERWC
00560000	00003000			PRV	-RW-	-RW-
00563000	00000000			MAP	-RW-	-RW-
00570000	00005000	Reserved (00560000)		MAP	-R--	-R--
00575000	0017B000	Reserved (00570000)		MAP	-R--	-R--
006F0000	00003000	Reserved (00570000)		MAP	-R--	-R--
006F3000	00005000	Reserved (00570000)		MAP	-R--	-R--
00700000	00181000			MAP	-R--	-R--
00890000	00049000			MAP	-R--	-R--
00899000	01387000	Reserved (00890000)		MAP	-R--	-R--
74740000	00008000			IMG	ERW-	ERWC
74750000	0005C000			IMG	-R--	ERWC
747B0000	0003F000			IMG	-R--	ERWC
759C0000	00001000	cryptbase.dll		IMG	-R--	ERWC
759C1000	00008000	"text"	Executable code	IMG	ER--	ERWC
759C9000	00001000	"data"	Initialized data	IMG	-RW-	ERWC
759CA000	00001000	"rsrc"	Resources	IMG	-R--	ERWC
759CB000	00001000	"reloc"	Base relocations	IMG	-R--	ERWC
759CD000	00001000	sspicli.dll		IMG	-R--	ERWC
759D1000	0000F000	Reserved (759CD000)		IMG	ER--	ERWC
759E0000	00016000	sspicli.dll		IMG	ER--	ERWC
759F6000	0000A000	Reserved (759E0000)		IMG	-R--	ERWC
75A00000	00001000	sspicli.dll		IMG	-RW-	ERWC
75A01000	0000F000	Reserved (75A00000)		IMG	-R--	ERWC
75A10000	00001000	sspicli.dll		IMG	-R--	ERWC
75A11000	0000F000	Reserved (75A10000)		IMG	-R--	ERWC
75A20000	00002000	sspicli.dll		IMG	-R--	ERWC
75A22000	0000E000	Reserved (75A20000)		IMG	-R--	ERWC
75B00000	00001000	msvcrt.dll		IMG	ER--	ERWC
75B01000	0009F000	"text"	Executable code	IMG	-RW-	ERWC
75BA0000	00007000	"data"	Initialized data	IMG	-RW-	ERWC
75BA7000	00001000	"rsrc"	Resources	IMG	-R--	ERWC
75BA8000	00004000	"reloc"	Base relocations	IMG	-R--	ERWC
75B00000	00000000	kernelbase.dll		IMG	-RW-	ERWC

After executing the call.



We noted now that permission for that region has been changed to read write here we can know that it is trying to write into this memory location.



Now here we again hit the VirtualProtect and now view the second parameter in memory map and now it has again changed its permission to read only.

The screenshot shows the x32dbg interface with the CPU window displaying assembly instructions. The instruction at address 75D0E339 is a call to `<kernelbase.VirtualProtectEx>`. The register window shows the state of registers: EAX=00000002, EBX=00001000, ECX=48F90000, EDI=00000000, ESP=0018FF50, ESI=00401000, and EIP=75D0E339. The memory map window shows the second parameter at address 0018FF50 with permissions RWX. The command window shows the command `CALL <kernelbase.VirtualProtectEx>`.

Just Before the call.

The screenshot shows the x32dbg interface with the Memory Map window. The memory map shows the second parameter at address 0018FF50 with permissions RWX. The command window shows the command `CALL <kernelbase.VirtualProtectEx>`.

Permission changed to read only after the call.

ramnitmgr.exe - PID: 1316 - Module: kernelbase.dll - Thread: Main Thread 1972 - x32dbg [Elevated]									
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)									
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace									
Address	Size	Info	Content	Type	Protection	Initial			
00010000	00010000			MAP	-RW---	-RW---			
00020000	00010000			PRV	-RW---	-RW---			
00030000	00010000			PRV	-RW---	-RW---			
00040000	00010000			IMG	-R---	ERW-C			
00050000	00039000	Reserved		PRV	-RW---	-RW---			
00089000	00007000			PRV	-RW-G	-RW---			
00090000	000FC000	Reserved		PRV	-RW---	-RW---			
0018C000	00004000	Thread 784 Stack		PRV	-RW-G	-RW---			
00190000	00004000			MAP	-R---	-R---			
001A0000	00001000			PRV	-RW---	-RW---			
001B0000	00006000			PRV	-RW---	-RW---			
001B6000	0007A000	Reserved (00180000)		PRV	-RW---	-RW---			
00230000	00007000			PRV	-RW---	-RW---			
00237000	000F9000	Reserved (00230000)		PRV	-RW---	-RW---			
00330000	00067000	\Device\HarddiskVolume1\windows\		MAP	-R---	-R---			
00400000	00001000	ramnitmgr.exe		IMG	-R---	ERW-C			
00401000	00034000	"UPX0"		IMG	ERW---	ERW-C			
00438000	00013000	"UPX1"		IMG	ERW---	ERW-C			
0044E000	0000F000	".rsr"		IMG	ERW-C	ERW-C			
00560000	00003000			PRV	-RW---	-RW---			
00563000	00000000	Reserved (00560000)		PRV	-RW---	-RW---			
00570000	00005000			MAP	-R---	-R---			
00575000	00178000	Reserved (00570000)		MAP	-R---	-R---			
006F0000	00003000			MAP	-R---	-R---			
006F3000	00005000	Reserved (00570000)		MAP	-R---	-R---			
00700000	00181000			MAP	-R---	-R---			
00890000	00049000			MAP	-R---	-R---			
00890000	01187000	Reserved (00890000)		MAP	-R---	-R---			
74740000	00008000			IMG	-R---	ERW-C			
74750000	0005C000			IMG	-R---	ERW-C			
74780000	0003F000			IMG	-R---	ERW-C			
759C0000	00001000	cryptbase.dll		IMG	-R---	ERW-C			
759C1000	00008000			IMG	-R---	ERW-C			
759C8000	00001000	".data"	Executable code	IMG	-RW---	ERW-C			
759CA000	00001000	".rsr"	Initialized data	IMG	-R---	ERW-C			
759CB000	00001000	".rsrc"	Resources	IMG	-R---	ERW-C			
759C8000	00001000	".reloc"	Base relocations	IMG	-R---	ERW-C			
759D0000	00001000	sspicli.dll		IMG	-R---	ERW-C			
759D1000	0000F000	Reserved (759D0000)		IMG	ER---	ERW-C			
759E0000	00016000	sspicli.dll		IMG	-RW---	ERW-C			
759F6000	0000A000	Reserved (759D0000)		IMG	-RW---	ERW-C			
75A00000	00001000	sspicli.dll		IMG	-RW---	ERW-C			
75A01000	0000F000	Reserved (759D0000)		IMG	-RW---	ERW-C			
75A10000	00001000	sspicli.dll		IMG	-R---	ERW-C			
75A11000	0000F000	Reserved (759D0000)		IMG	-R---	ERW-C			
75A20000	00002000	sspicli.dll		IMG	-R---	ERW-C			
75A22000	0000E000	Reserved (759D0000)		IMG	-R---	ERW-C			
75B00000	00001000	msvcrt.dll		IMG	-R---	ERW-C			

Now we hit breakpoint at VirtualAlloc

ramnitmgr.exe - PID: 1316 - Module: kernel32.dll - Thread: Main Thread 1972 - x32dbg [Elevated]

File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)

CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

75F21856 8BFF mov edi,edi
75F21858 55 push ebp
75F21859 8BEC mov ebp,esp
75F2185B 5D pop ebp
75F2185C EB 05 jmp <JMP,&VirtualAlloc>
75F2185E 90 nop
75F2185F 90 nop
75F21860 90 nop
75F21861 90 nop
75F21862 90 nop
75F21863 FF25 0809F275 jmp dword ptr ds:[&VirtualAlloc]
75F21869 90 nop
75F2186A 90 nop
75F2186B 90 nop
75F2186C 90 nop
75F2186D 90 nop
75F2186E 8BFF mov edi,edi
75F21870 55 push ebp
75F21871 8BEC mov ebp,esp
75F21873 5D pop ebp
75F21874 EB 05 jmp <JMP,&VirtualFree>
75F21876 90 nop
75F21877 90 nop
75F21878 90 nop
75F21879 90 nop
75F2187A 90 nop
75F2187B FF25 1009F275 jmp dword ptr ds:[&VirtualFree]
75F21881 90 nop
75F21882 90 nop
75F21883 90 nop
75F21884 90 nop
!!!</p><p>edi=F26046AE</p><p>.text:75F21856 kernel32.dll:11856 #1C56 <VirtualAlloc></p><p>Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct</p><p>0018FF74 00401F9F return to ramnitmgr.00401F9F</p><p>Address Hex ASCII</p><p>00400000 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZ.....YY.....</p></div>

After this call the allocated memory will be shown in eax register.

ramnitmgr.exe - PID: 1316 - Module: kernelbase.dll - Thread: Main Thread 1972 - x32dbg [Elevated]

File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)

CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

75D0E365 8BFF push ebp
75D0E367 55 mov ebp, esp
75D0E368 8BEC mov ebp, esp
75D0E36A FF75 14 push dword ptr ss:[ebp+14]
75D0E36D FF75 10 push dword ptr ss:[ebp+10]
75D0E370 FF75 0C push dword ptr ss:[ebp+C]
75D0E373 FF75 08 push dword ptr ss:[ebp+8]
75D0E376 6A FF push 57
75D0E378 E8 4BFFFFFF call <kernelbase.VirtualAllocEx>
75D0E37D 5D pop ebp
75D0E37E C2 1000 ret 10
75D0E381 CC int3
75D0E382 CC int3
75D0E383 CC int3
75D0E384 CC int3
75D0E385 CC int3
75D0E386 8BFF mov edi, edi
75D0E388 55 push ebp
75D0E389 8BEC mov ebp, esp
75D0E38B 56 push esi
75D0E38C 8B75 0C mov esi, dword ptr ss:[ebp+C]
75D0E38F 85F6 test esi, esi
75D0E391 75 0C jne kernelbase.75D0E39F
75D0E393 6A 57 push 57
75D0E395 FF15 4C100075 call dword ptr ds:[<RtlRestoreLastWin32Error>]
75D0E398 33C0 xor eax, eax
75D0E399 jmp kernelbase.75D0E306
75D0E39B 44 37 lea eax, dword ptr ss:[ebp+C]
75D0E39D 50 push eax
75D0E39F FF15 E011D075 call dword ptr ds:[<RtlGetCurrentProcessId>]
75D0E3A2 56 push esi
75D0E3A3 int3
75D0E3A9 int3

VirtualAllocEx

GetLogicalProcessorInformation

EAX 0044200C <ram
ECX 75D0E378
EDX 00000057 'm'
EBP 0018FF70
ESP 0018FF5C
ESI F9302355
EDI F26046AE

EIP 75D0E378 kern

EFLAGS 00000246
ZF 1 PF 1 AF 0
OF 0 SF 0 DF 0
CF 0 TF 0 IF 1

LastError 00000057 (ER
LastStatus C00000F2 (ST

GS 002B FS 0053
ES 002B DS 002B
CS 0023 SS 002B

Default (stdcall) 5

1: [esp] FFFFFFFF
2: [esp+4] 00000000
3: [esp+8] 00001000
4: [esp+C] 00001000
5: [esp+10] 00000040

Address Hex
00400000 40 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZ.....YY..
00400010 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00@.....
00400020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00400030 00 00 00 00 00 00 00 00 00 00 00 00 88 00 00 00
0018FF5C FFFFFFFF
0018FF60 00000000
0018FF64 00001000
0018FF68 00001000
0018FF6C 00000040
0018FF70 0018FF94
0018FF74 00401F9F return to ramnitmgr.00401F9F from ???

Here we can see the allocated memory is at 3A0000 and dump this location at dump1.

ramnitmgr.exe - PID: 1316 - Module: kernelbase.dll - Thread: Main Thread 1972 - x32dbg [Elevated]

File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)

CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

75D0E365 8BFF push ebp
75D0E367 55 mov ebp, esp
75D0E368 8BEC mov ebp, esp
75D0E36A FF75 14 push dword ptr ss:[ebp+14]
75D0E36D FF75 10 push dword ptr ss:[ebp+10]
75D0E370 FF75 0C push dword ptr ss:[ebp+C]
75D0E373 FF75 08 push dword ptr ss:[ebp+8]
75D0E376 6A FF push 57
75D0E378 E8 4BFFFFFF call <kernelbase.VirtualAllocEx>
75D0E37D 5D pop ebp
75D0E37E C2 1000 ret 10
75D0E381 CC int3
75D0E382 CC int3
75D0E383 CC int3
75D0E384 CC int3
75D0E385 CC int3
75D0E386 8BFF mov edi, edi
75D0E388 55 push ebp
75D0E389 8BEC mov ebp, esp
75D0E38B 56 push esi
75D0E38C 8B75 0C mov esi, dword ptr ss:[ebp+C]
75D0E38F 85F6 test esi, esi
75D0E391 75 0C jne kernelbase.75D0E39F
75D0E393 6A 57 push 57
75D0E395 FF15 4C100075 call dword ptr ds:[<RtlRestoreLastWin32Error>]
75D0E398 33C0 xor eax, eax
75D0E399 jmp kernelbase.75D0E306
75D0E39B 44 37 lea eax, dword ptr ss:[ebp+C]
75D0E39D 50 push eax
75D0E39F FF15 E011D075 call dword ptr ds:[<RtlGetCurrentProcessId>]
75D0E3A2 56 push esi
75D0E3A3 int3
75D0E3A9 int3

VirtualAllocEx

GetLogicalProcessorInformation

EAX 003A0000
ECX 48F90000
EDX 0008E3C8
EBP 0018FF70
ESP 0018FF70
ESI F9302355
EDI F26046AE

EIP 75D0E37D kern

EFLAGS 00000244
ZF 1 PF 1 AF 0
OF 0 SF 0 DF 0
CF 0 TF 0 IF 1

LastError 000000
LastStatus C00000

GS 002B FS 0053
ES 002B DS 002B
CS 0023 SS 002B

Default (stdcall) 5

1: [esp+4] 00401F9F
2: [esp+8] 00000000
3: [esp+C] 00001000
4: [esp+10] 00001000
5: [esp+14] 00000040

Address Hex
003A0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003A0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003A0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003A0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003A0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003A0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0018FF74 00401F9F return to ramnitmgr.00401F9F from ???
0018FF78 00000000
0018FF7C 00001000
0018FF80 00001000
0018FF84 00000040
0018FF88 0040141F return to ramnitmgr.0040141F from ram
0018FF8C 75F233CA return to kernel32.75F233CA from ???
0018FF90 75F0E000
0018FF94 75F0E000

[illegible]

ramnitmgr.exe - PID: 1316 - Module: kernelbase.dll - Thread: Main Thread 1972 - x32dbg [Elevated]

File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)

CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

75D0E365 8BFF push edi
 75D0E367 55 push ebp
 75D0E368 8BEC mov ebp,esp
 75D0E36A FF75 14 push dword ptr ss:[ebp+14]
 75D0E36D FF75 10 push dword ptr ss:[ebp+10]
 75D0E370 FF75 0C push dword ptr ss:[ebp+8]
 75D0E373 FF75 08 push dword ptr ss:[ebp+4]
 75D0E376 6A FF push ffffffff
 75D0E37D E8 4BFFFFFF call kkernelbase.VirtualAllocEx
 75D0E37E 8BEB mov ebx,ebp
 75D0E381 C2 1000 ret 10
 75D0E382 CC int3
 75D0E383 CC int3
 75D0E384 CC int3
 75D0E385 CC int3
 75D0E386 8BFF mov edi,edi
 75D0E388 55 push ebp
 75D0E389 8BEC mov ebp,esp
 75D0E38B 56 mov esi,dword ptr ss:[ebp+6]
 75D0E38C 8B75 0C mov esi,dword ptr ss:[ebp+C]
 75D0E38F 85F6 test esi,esi
 75D0E391 75 0C jnz kernelbase.75D0E39F
 75D0E393 6A 57 push 57
 75D0E395 FF15 4C10D075 call dword ptr ds:[<RtRestoreLastWin32...>
 75D0E398 33C0 xor eax,ebx
 75D0E399 EB 37 jmp kernelbase.75D0E3D6
 75D0E39F 8D45 0C lea eax,dword ptr ss:[ebp+C]
 75D0E3A2 50 push eax
 75D0E3A3 5F call dword ptr ds:[<RtGetCurrentProce...>
 75D0E3A9 56 push esi

VirtualAlloc

GetLogicalProcessorInformation

ebp=0018F344

.text:75D0E37D kernelbase.dll:5E37D #077D

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

Address	Hex	ASCII
00380000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00380010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00380020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00380030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00380040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00380050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00380060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0018F344 0018F3BC return to 003A0984 from ???

0018F3A8 003A0984
 0018F3AC 00000000
 0018F3B0 00008000
 0018F3B4 00003000
 0018F3B8 00005040
 0018F3BC 0018FF88
 0018F3C0 003A0534 return to 003A0534 from 003A05D0
 0018F3C4 00000000
 0018F3C8 00008000

Hide

EAX 00380000
 EAX FFFBFEB0
 ECX 4BF90000
 EDI 0000E3C9
 EBP 0018F344
 ESP 0018F344
 ESI 00061FF8
 EDI 003A0B30
 EIP 75D0E37D

EFlags 00000244
 ZF 1 PF 1 AF 0
 OF 0 SF 0 DF 0
 CF 0 TF 0 IF 1

LastError 0000007F
 LastStatus 00000000

GS 002B F5 0053
 ES 002B D5 0028
 CS 0023 55 0028

Default (stdcall)

1: [esp+4] 003A0984
 2: [esp+8] 00000000
 3: [esp+C] 00008000
 4: [esp+10] 00003000
 5: [esp+14] 00000040

The screenshot shows the Immunity Debugger interface with the following details:

- Menu Bar:** File, View, Debug, Tracing, Plugins, Favourites, Options, Help, Jan 13 2022 (TitanEngine)
- Toolbar:** Includes icons for CPU, Log, Notes, Breakpoints, Memory Map, Call Stack, SEH, Script, Symbols, Source, References, Threads, Handles, and Trace.
- CPU Window:**
 - Address: 75F21856
 - Disassembly:


```

            75F21856 55      push    ebp
            75F21857 5D      pop     ebp
            75F21858 EB 05    jmp     <MP.>VirtualAlloc
            75F21859 90      nop
            75F2185A 90      nop
            75F2185B 90      nop
            75F2185C 90      nop
            75F2185D 90      nop
            75F2185E 90      nop
            75F2185F 90      nop
            75F21860 90      nop
            75F21861 90      nop
            75F21862 90      nop
            75F21863 FF25 0809275 jmp     dword ptr ds:[<VirtualAlloc>]
            75F21864 90      nop
            75F21865 90      nop
            75F21866 90      nop
            75F21867 90      nop
            75F21868 90      nop
            75F21869 90      nop
            75F2186A 90      nop
            75F2186B 90      nop
            75F2186C 90      nop
            75F2186D 90      nop
            75F2186E 90      nop
            75F2186F 90      nop
            75F21870 55      push    edi
            75F21871 5D      pop     edi
            75F21872 EB 05    jmp     <MP.>VirtualFree
            75F21873 90      nop
            75F21874 90      nop
            75F21875 90      nop
            75F21876 90      nop
            75F21877 90      nop
            75F21878 90      nop
            75F21879 90      nop
            75F2187A 90      nop
            75F2187B FF25 1009275 jmp     dword ptr ds:[<VirtualFree>]
            75F2187C 90      nop
            75F2187D 90      nop
            75F2187E 90      nop
            75F2187F 90      nop
            75F21880 90      nop
          
```
- Registers Window:**
 - EAX: 75F21856
 - ECX: FFFEBE00
 - EDX: 75F10000
 - EBX: 0018F38C
 - ESP: 0018F3A8
 - ESI: 00061F58
 - EDI: 003A0B3D
 - EIP: 75F21856
 - EFLAGS: 00000304
 - ZF: 0, PF: 1, AF: 0
 - OF: 0, SF: 0, DF: 0
 - CF: 0, TF: 1, IF: 1
 - LastError: 0000007F
 - LastStatus: 00000000
 - GS: 0028, FS: 0013
 - ES: 0028, DS: 0028
 - CS: 0023, SS: 0028
- Stack Window:**
 - Default (stdcall):
 - [esp+4] 00000000
 - [esp+8] 0001C832
 - [esp+C] 00003000
 - [esp+10] 00000004
 - [esp+14] 00018F58
- Command Window:**

```

.edi=003A0B3D

.text:75F21856 kernel32.dll:11856 #C56 <VirtualAlloc>
  
```
- Bottom Panel:**
 - Buttons: Dump 1, Dump 2, Dump 3, Dump 4, Dump 5, Watch 1, [x]=Locals, Struct
 - Memory dump table with columns: Address, Hex, ASCII.
 - Search results for 0018F3A8 and 003A0984, showing return addresses.

ramming7.exe - PID: 1316 - Module: kernelbase.dll - Thread: Main Thread 1972 - x32dbg (Elevated)

File View Debug Tracing Plugins Favourites Open Help Jan 13 2022 (TitanEngine)

CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

75D0E365 8BF5 mov edi,edi
 75D0E367 55 push ebp
 75D0E368 8BEC mov ebp,esp
 75D0E373 7F75 14 push dword ptr ss:[ebp+14]
 75D0E380 7F75 10 push dword ptr ss:[ebp+10]
 75D0E370 7F75 0C push dword ptr ss:[ebp+C]
 75D0E373 7F75 08 push dword ptr ss:[ebp+8]
 75D0E376 6A FF push ffffffff
 75D0E378 call <kernelbase.VirtualAlloc>
 75D0E37D 50 push ebp
 75D0E37E C2 1000 ret 10
 75D0E381 CC int3
 75D0E382 CC int3
 75D0E383 CC int3
 75D0E384 CC int3
 75D0E385 CC int3
 75D0E386 8BF5 mov edi,edi
 75D0E388 55 push ebp
 75D0E389 8BEC mov ebp,esp
 75D0E38B 56 push esi
 75D0E38C 8B75 0C mov esi,dword ptr ss:[ebp+C]
 75D0E38F 55F6 test esi,esi
 75D0E391 75 0C jnz kernelbase.75D0E39F
 75D0E393 6A 57 push 57
 75D0E395 75D0E395 7F75 4C0075 call dword ptr ds:[<Kd!RestoreLastWin32>
 75D0E398 33C0 xor eax,eax
 75D0E39D EB 37 jmp kernelbase.75D0E3A8
 75D0E39F 7F75 0C jnz kernelbase.75D0E3A8
 75D0E3A2 50 push eax
 75D0E3A3 FF75 0C call dword ptr ds:[<Kd!GetCurrentProcess>
 75D0E3A9 56 push esi
 75D0E3AA 55
 75D0E3AB 56
 75D0E3AC 57
 75D0E3AD 58
 75D0E3AE 59
 75D0E3AF 5A
 75D0E3B0 5B
 75D0E3B1 5C
 75D0E3B2 5D
 75D0E3B3 5E
 75D0E3B4 5F
 75D0E3B5 60
 75D0E3B6 61
 75D0E3B7 62
 75D0E3B8 63
 75D0E3B9 64
 75D0E3BA 65
 75D0E3BB 66
 75D0E3BC 67
 75D0E3BD 68
 75D0E3BE 69
 75D0E3BF 6A
 75D0E3C0 6B
 75D0E3C1 6C
 75D0E3C2 6D
 75D0E3C3 6E
 75D0E3C4 6F
 75D0E3C5 70
 75D0E3C6 71
 75D0E3C7 72
 75D0E3C8 73
 75D0E3C9 74
 75D0E3CA 75
 75D0E3CB 76
 75D0E3CC 77
 75D0E3CD 78
 75D0E3CE 79
 75D0E3CF 7A
 75D0E3D0 7B
 75D0E3D1 7C
 75D0E3D2 7D
 75D0E3D3 7E
 75D0E3D4 7F
 75D0E3D5 80
 75D0E3D6 81
 75D0E3D7 82
 75D0E3D8 83
 75D0E3D9 84
 75D0E3DA 85
 75D0E3DB 86
 75D0E3DC 87
 75D0E3DD 88
 75D0E3DE 89
 75D0E3DF 8A
 75D0E3E0 8B
 75D0E3E1 8C
 75D0E3E2 8D
 75D0E3E3 8E
 75D0E3E4 8F
 75D0E3E5 90
 75D0E3E6 91
 75D0E3E7 92
 75D0E3E8 93
 75D0E3E9 94
 75D0E3EA 95
 75D0E3EB 96
 75D0E3EC 97
 75D0E3ED 98
 75D0E3EE 99
 75D0E3EF 9A
 75D0E3F0 9B
 75D0E3F1 9C
 75D0E3F2 9D
 75D0E3F3 9E
 75D0E3F4 9F
 75D0E3F5 A0
 75D0E3F6 A1
 75D0E3F7 A2
 75D0E3F8 A3
 75D0E3F9 A4
 75D0E3FA A5
 75D0E3FB A6
 75D0E3FC A7
 75D0E3FD A8
 75D0E3FE A9
 75D0E3FF AA
 75D0E400 AB
 75D0E401 AC
 75D0E402 AD
 75D0E403 AE
 75D0E404 AF
 75D0E405 B0
 75D0E406 B1
 75D0E407 B2
 75D0E408 B3
 75D0E409 B4
 75D0E40A B5
 75D0E40B B6
 75D0E40C B7
 75D0E40D B8
 75D0E40E B9
 75D0E40F BA
 75D0E410 BB
 75D0E411 BC
 75D0E412 BD
 75D0E413 BE
 75D0E414 BF
 75D0E415 C0
 75D0E416 C1
 75D0E417 C2
 75D0E418 C3
 75D0E419 C4
 75D0E41A C5
 75D0E41B C6
 75D0E41C C7
 75D0E41D C8
 75D0E41E C9
 75D0E41F CA
 75D0E420 CB
 75D0E421 CC
 75D0E422 CD
 75D0E423 CE
 75D0E424 CF
 75D0E425 D0
 75D0E426 D1
 75D0E427 D2
 75D0E428 D3
 75D0E429 D4
 75D0E42A D5
 75D0E42B D6
 75D0E42C D7
 75D0E42D D8
 75D0E42E D9
 75D0E42F DA
 75D0E430 DB
 75D0E431 DC
 75D0E432 DD
 75D0E433 DE
 75D0E434 DF
 75D0E435 E0
 75D0E436 E1
 75D0E437 E2
 75D0E438 E3
 75D0E439 E4
 75D0E43A E5
 75D0E43B E6
 75D0E43C E7
 75D0E43D E8
 75D0E43E E9
 75D0E43F EA
 75D0E440 EB
 75D0E441 EC
 75D0E442 ED
 75D0E443 EE
 75D0E444 EF
 75D0E445 F0
 75D0E446 F1
 75D0E447 F2
 75D0E448 F3
 75D0E449 F4
 75D0E44A F5
 75D0E44B F6
 75D0E44C F7
 75D0E44D F8
 75D0E44E F9
 75D0E44F FA
 75D0E450 FB
 75D0E451 FC
 75D0E452 FD
 75D0E453 FE
 75D0E454 FF

VirtualAlloc

GetLogicalProcessorInformation

kernelbase.VirtualAlloc

kernelbase.dll:75D0E378 kernelbase.dll:75D0E378

Hide FPU

EAX 75F21856 <kernel32.75F21856>
 ECX FF8FEBE0 <kernel32.75F21856>
 EDI 00000000 <kernel32.75F21856>
 ESI 0018F340 <kernel32.75F21856>
 ESP 0018F340 <kernel32.75F21856>
 EIP 0018F340 <kernel32.75F21856>
 EDI 003A0B30 <kernel32.75F21856>
 EIP 75D0E378 kernelbase.dll:75D0E378
 EFLAGS 00000206
 ZF 0 PF 1 AF 0
 OF 0 SF 0 DF 0
 CF 0 TF 0 IF 1
 LastError 0000007E (ERROR_INVALID_PARAMETER)
 LastStatus 00000000 (STATUS_SUCCESS)

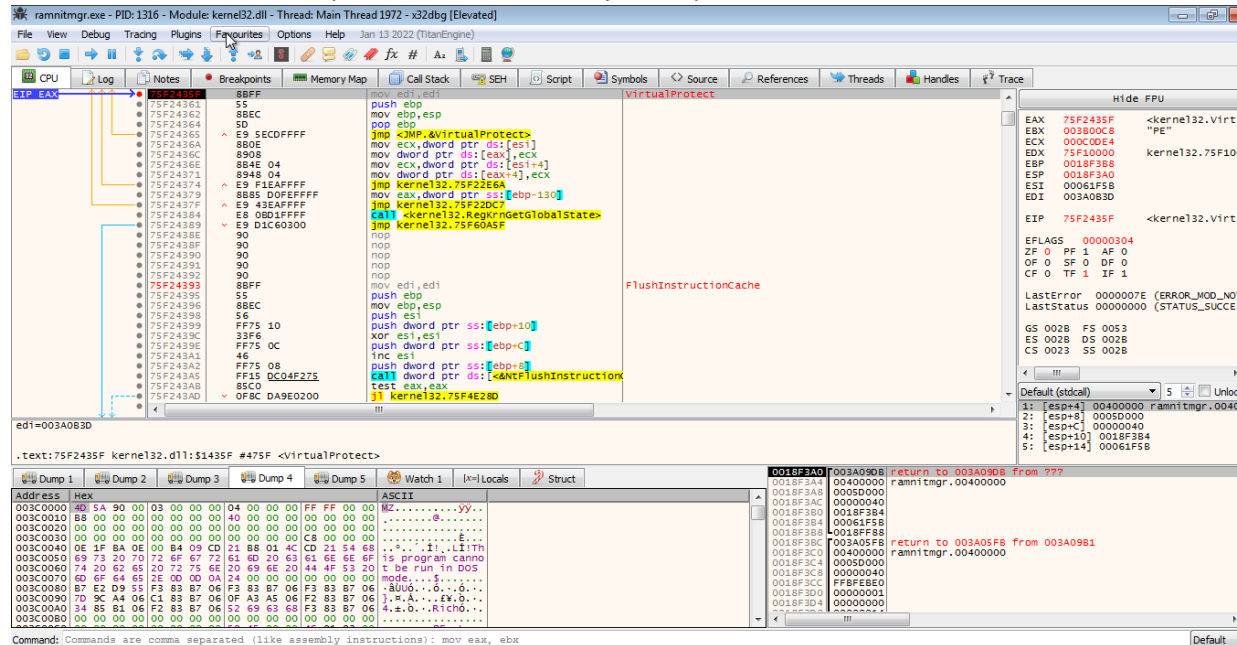
GS 0028 F5 0053
 ES 0028 D5 0028
 CS 0028 55 0028

Default (stdcall) 5
 1: [esp] ffffffff
 2: [esp+1] 00000000
 3: [esp+8] 0001C832
 4: [esp+C] 00003000
 5: [esp+10] 00000040

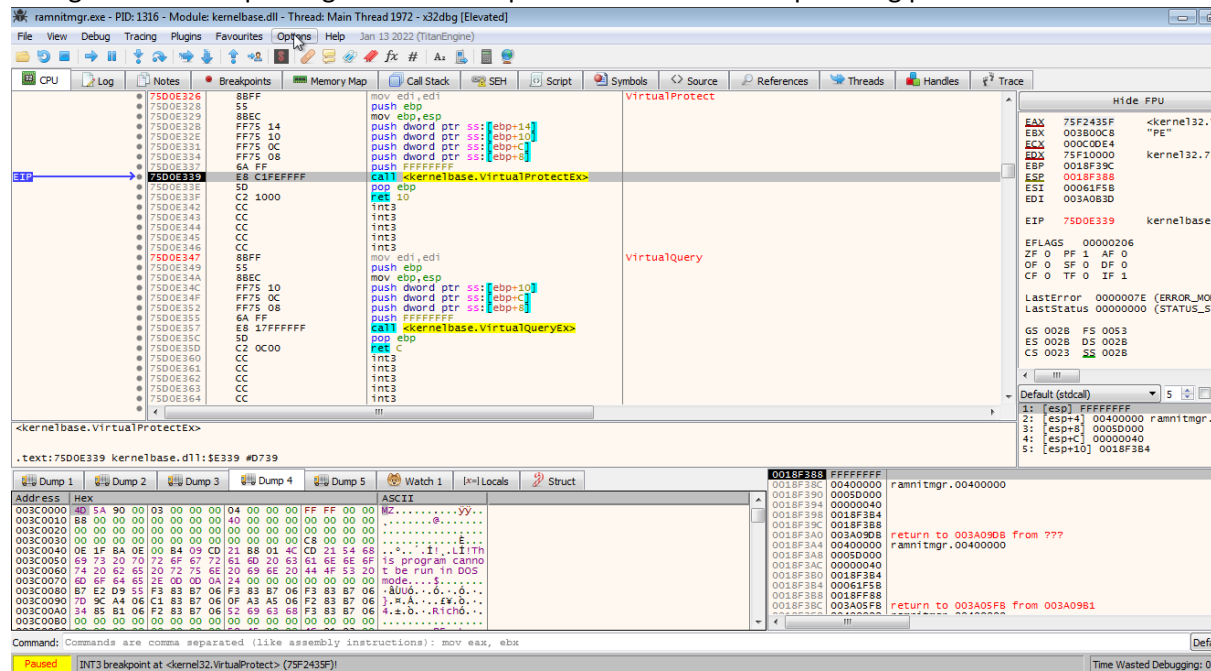
0018F390 ffffffff
 0018F398 00000000
 0018F39C 0001C832
 0018F3A0 0

The screenshot shows the Immunity Debugger interface. The CPU window displays assembly instructions for the kernelbase.dll module, starting at address 75D0E370. The instruction 'call kernelbase.VirtualA1ToCEx' is highlighted. The right pane shows the 'VirtualA1ToC' function's stack frame, with registers EAX, ECX, EDI, and EIP. The bottom pane shows a memory dump of the kernelbase.dll module, with the address 0018F3A4 highlighted.

Now here we hit the breakpoint at VirtualProtect just step over it.



Now here note the second parameter address that is the region whose permission bits will be changed. Now as unpacking has been completed now we will dump it using process hacker.



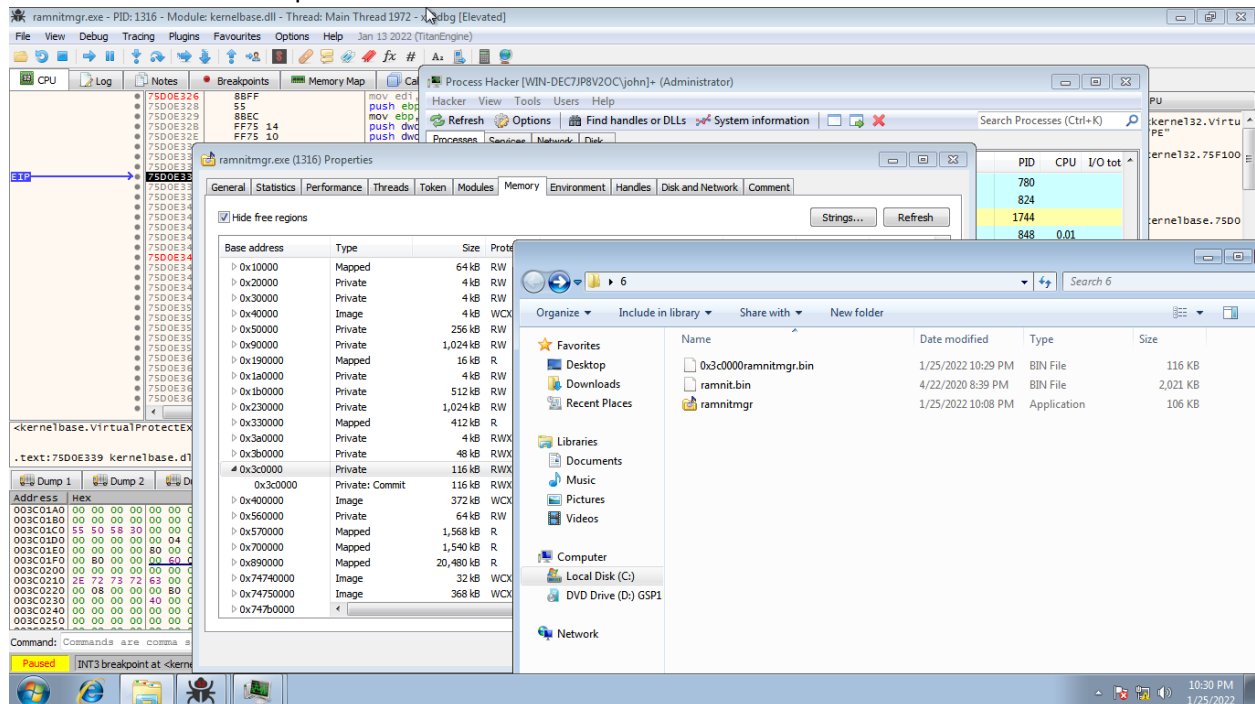
The screenshot displays a Windows 10 desktop with three applications open:

- RAMMap (ramntmgr.exe - PID: 1316):** Shows memory usage for the process. The 'Commit' column is highlighted in red, indicating a large memory commitment of 1.22 GB (1752 MB) for the private space (0x3c0000 - 0x3dd000).
- Process Hacker (Process Hacker [WIN-DEC7/P8V2OC\john] - Administrator):** Shows the process 'ramntmgr.exe (1316)' with a memory usage of 0.01 GB (848 MB).
- Task Manager:** Shows the process 'ramntmgr.exe' with a memory usage of 1.22 GB (1752 MB).

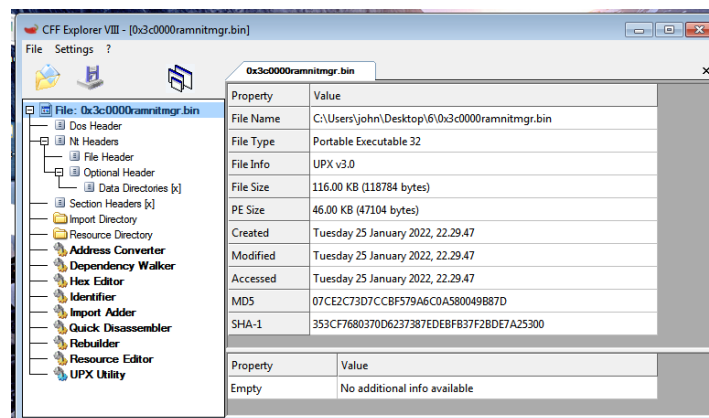
The desktop background is a blue and white geometric pattern.

[illegible]

Now here we have dumped the file.



Now here we are using CFF Explorer to unpack it and here it has recognised that it is packed with UPX v3.0.



Now click on UPX Utility tab and click on unpack button and here it had successfully unpacked it now save it.

