Unpacking Dridex malware

muha2xmad.github.io/unpacking/dridex/

January 11, 2022



Muhammad Hasan Ali

Malware Analysis learner

2 minute read

As-salamu Alaykum

Introducton

Dridex is a famous banking Trojan which appeared around 2011 and is still very active today. This is because of its evolution and its complex architecture, which is based on proxy layers to hide the main command and control servers (C&C). The APT known as TA505 is associated to Dridex. $\underline{1}$

MD5: 6A8401448A5BD2B540850F811B20A66D

Static

Open it in **DiE** to see **Entropy**. It's too high and if we open the sample in **IDA**. **IDA** will pop up a warning which means that the sample is packed. After it's opened in **IDA** we see that it's less number of functions and no imports.



Figure(1):

Unpacking Process

Then open the sample with x32dbg and set 3 breakpoints VirtualAlloc and VirtualProtect and CreateProcessInternalW. Then press F9 to hit the first BP. We hit VirtualAlloc then Execute till return and dump EAX. We see that's empty. Then run again and Execute till return and dump EAX. Then run again for the third time and Execute till return and dump EAX. We notice that there's a PE header starts with MZ magic byte which we can assume that's our unpacked file. **Don't close** x32dbg yet Follow in memory map and save it then open it in pestudio. It shows that we didn't finish the unpacking process.



Figure(2):

We keep doing the same steps till we hit VirtualProtect BP. then Execute till retrun then we look at the second parameter which will change the permission at this memory location which is 10000000.



Figure(3):

If you follow in Memory Map you will see RWC which means that it still writing which still unpacking.

01D70000	0034F000			MAP	-R	-R
10000000	00001000	dridex.exe		IMG	-RWC-	ERWC-
10001000	00003000	".text"	Executable code	IMG	-KWC-	ERWC-
10004000	00001000	".rdata"	Read-only initialized data	a IMG	-RW	ERWC-
10005000	00002000	".data"	Initialized data	IMG	-RW	ERWC-
10007000	0002C000	".text1"	Executable code	IMG	-RWC-	ERWC-
10033000	00001000	".rsrc"	Resources	IMG	-RWC-	ERWC-
10034000	00001000	".reloc"	Base relocations	IMG	-RWC-	ERWC-

Figure(4):

Then run again which will hit VirtualProtect again. Then Execute till return and it points at the same location 10000000, but this time shows us it's only R in memory map. Now we don't know that if it's done or not. So run again and Execute till return we see it points at another location but in the same region of memory of 10000000 which is 10001000.



Figure(5):

This the 3rd time to hit VirtualProtect BP we keep doing the same steps which will point at another locations but in the same region of memory of 10000000. and in the 6th one we **stop**. We press F9 to run. It shows us **EXCEPTION_BREAKPOINT** it's done unpacking. **Don't close the debugger**.

🕮 CPU 🍃 Log 🗊 Notes 🔹 Breakpoints 🚥 Memory Map	🗐 Call Stack 👒 SEH 🔟 Script 🔮	Symbols 🗢 Source 🏓 References 👒	🕨 Threads 🔒 Handles 🛛 Trace			
1001A43A A3 64820210	mov dword ptr ds:[<&RtlDestroyHeap>],e					Hide FPU
1001.4441 G 4 00 1001.4443 G 4 00 1001.4443 G 4 00 1001.4445 G 4 00 1001.4445 G 4 00 1001.4447 G 4 00 1001.4446 F 60 1001.4446 A 3 80820210 	push o push o push o push o push o call esi call esi mov dword ptr ds:[10028280],eax pop_esi pop_esi	10028280:&"9¥?"¶ " esi:'₩2"		E. E. E. E. E. E. E. E. E.	AX 7773E106 BX 00000104 CX 02240000 DX 00010240 BP 0018FD8C SP 0018FBDC SI 77280000 DI 00000000	<ntdll.rtlallocateheap> ↓ A' "94?" ¶ " &"MY!x18" &"97?" ¶ " MZ"</ntdll.rtlallocateheap>
- 1001A454, 68 5000805 - 1001A454, 68 5000805 - 1001A454, 85 50 50257 	push E259A026 push 238D163 call dridex.10000000 test eax.eax je dridex.1001A476 mov ecx.dword ptr ds:[10028280] push ebx push 8			E: Ef Zr Or C	IP 1001A473 FLAGS 00000206 F 0 PF 1 AF 0 F 0 SF 0 DF 0 F 0 TF 0 IF 1	
1001A471 51 1001A472 CC	push ecx	ecx:"9¥?"¶ "			astError 00000057 astStatus C000000	7 (ERROR_INVALID_PARAMETER) D (STATUS_INVALID_PARAMETER)
10014474 58 10014475 C3 10014475 C3 10014476 33C0 10014478 58 10014479 C3	pop ebx ret xor eax,eax pop ebx iet-3			G: E: C:	S 0028 FS 0053 S 0028 DS 0028 S 0023 <u>SS</u> 0028	
1001A47B CC 1001A47C CC 1001A47C CC				De T 1:	efault (stdcall) [esp+4]_0000008	
.text1:1001A473 dridex.exe:\$1A473 #19473		<u> </u>		► 2: 3: 4: 5:	[esp+8] 00000104 [esp+C] 418EE2AE [esp+10] 1000B8C [esp+14] 0000010	F dridex.1000B8CF
🖷 Dump 1 🖷 Dump 2 🕮 Dump 3 🕮 Dump 4 🕮 Dump 5	Watch 1 Locals 2 Struct		0018FBDC 02240000 "9¥?"1 "			
Address Hex Ascr 11 0200000000000000000000000000000000000			00187856 406001104 00187856 4185524 00187856 40900104 00188785 00900104 00188750 00900104 00188750 4185524 00188750 415544410 00187505 415544410 00187505 415544410			•
command: Commands are comma separated (like assembly instruction	ns): mov eax, ebx					Default 👻
Paused First chance exception on 1001A472 (80000003, EXCEPTION_BREAKPOIN	D (2)					Time Wasted Debugging: 0:07:53:22

Figure(6):

Now we hit **VirtualAlloc** 3 times and **VirtualProtect** 6 times at the 6th one we finished unpacking. We will restore the snapshot and do it again till hit **VirtualProtect** at the 6th time and **Execute till return**.

Then open **Process Hacker** tool to to save the unpacked file.

📮 Process Hacker (1 🚽 🔤 🔤 🔤 🔤 🔤 🔤											
Hacker View Tools Users Help								Modules			
😂 Refresh 👳 Options 🛛 🏭 Find handles or DLLs 🛛 🚈 System information 🗌 📃						Memory	Envir	onment Hand	dles	PU	Comment
Processes Services Network Disk						💉 🍈 🗖					
Name	PID	CPU	I/O total r	Private		✓ Hide free regions Strings			Strings	Refresh	
spoolsv.exe	260			5.96 2		Race address		Turpe	Cir	o Brotaction	
svchost.exe	1036			8.2 MB		base autress		Privato	512		Hoap (1
svchost.exe	1180	0.03	552 B/s	12.05 MB		> 0x3d0000		Manned	1.568	BR	neap (i
taskhost.exe	1188	0.01		11.54 MB		> 0x570000		Private	1,024	B RW	Heap 3
svchost.exe	1292			3.63 MB		⊳ 0x670000		Mapped	1,540	BR	
VGAuthService.exe	1472			2.42 MB		⊳ 0x800000		Mapped	20,480	BR	
✓ I wm3dservice exe	1644			1.2 MB		▷ 0x1c00000		Private	168	B RW	
wm3dservice.exe	1664			1 34 MB		▷ 0x1d30000		Private	256	B RW	Heap 3
	1676	0.49	2 02 kB/c	11.87 MB		▷ 0x1d70000		Mapped	3,388	BR	=
	1906	0.45	2.02 KD/3			▷ 0x2240000		Private	256	B RW	Heap 3
sypsvc.exe	2004			3		■ 0x10000000		Read/Write mem	norv		C:\Usei
svchost.exe	2004			1.54 MD		0x100000	00 F	Save			C:\Use
svchost.exe	2109			1.34 IVID		0x100220	00 L	Change protection			C:\Usei
dinost.exe	2100			4.07 MB		0x100280	00		11		C:\Usei
se msatc.exe	2288			3.29 MB		0x100290	00	Free			C:\Usei
SearchIndexer.exe	2436			27.34 MB		0x1002a0	00	Decommit			C:\Usei
Isass.exe	476	0.15		3.55 MB		▷ 0x724a0000		Read/Write addre	ess	¢	C:\Win
Ism.exe	484			2.18 MB		▷ 0x72d30000		Copy	Ct	I+C	C:\Win
Csrss.exe	368	0.47	1.01 kB/s	8.39 MB		▷ 0x748b0000		Copy "Base addre	ss"	¢	C:\Win
🔛 winlogon.exe	404			2.77 MB		▷ 0x748c0000		The second secon	050		C:\Win
4 🧊 explorer.exe	2716	0.19		39.05 MB		> 0x/4920000		Image	252	B WCX	C:\Win
vmtoolsd.exe	2892	0.31	1.07 kB/s	10.03 MB		> 0x74e70000		Image	394	B WCX	C:\Win
▲ ╬ x32dbg.exe	2496	0.43		42.49 MB		> 0x74ef0000		Image	72	B WCX	C:\Win _
🔝 dridex.exe 🚤 🚺	2724	0.02		1.46 MB		> 0x74f10000		4	72	5 110/	
ProcessHacker.exe	2880	2.57		9.97 MB				•			P
CPU Usage: 11.41% Physical memory:	773.28 MB	(37.77%	6) Processes:	38							Close

Figure(7):

Unmapping

Then we repair the unapcked file to restore the imports table. As we did <u>Here</u> and save it. Then open it in **pestudio** if you unmapped it right, will show the imports table and you could analyze it easy.

Article quote

النعيمُ لا يُدرك بالنعيم

REF

1- https://cyber-anubis.github.io/malware%20analysis/dridex/