DEFENDING WORLD SECURITY



Disass : a new malware analysis framework

Ivan Fontarensky

AN EADS COMPANY

Who am I?

- Ivan Fontarensky

 © ifontarensky
- Working in CSIRT Team for Cassidian CyberSecurity
 - Incident Response
 - Malware Analysis
- Have been "playing around" with malware analysis
- Contributing on other project : Yara Community



Reason why we build Disass?

- Cassidian CyberSecurity is involved in the "ACDC" European project.
- We found piles of malware during Incident Response
 Need to quickly extract valuable information from the malware
- Malware streams received from partners
 - Analysis must be automated
- Malware are evolving fast
 - Building basic analysis scripts is not enough



What is Disass?

- Framework to ease reverse automation
- Written in python 2.7
- Licensed under GPL v3
- Disass is based on :
 - Distorm3 (linear disassembly engine) by Gil Dabah
 - pefile by Ero Carrera





- Disass is not another disassembler
- Disass is not a debugger
- Disass is not an emulator



Main functionalities

- Automated disassembly engine
 - Human readable automation scripts
 - Interactive shell to help writing automation script
- Evaluate a possible register value
- Follow both branches in case of conditional jump
- Jumping in the middle of opcodes is allowed by Distorm3

=> Resilience to malware evolution



We want to get the Mutex value for this malware :

szusername= byte ptr -&4n var_4= dword ptr -4 argc= dword ptr 4 argv= dword ptr 8 envp= dword ptr 0Ch
sub esp, 184h
mov eax,security_cookie
xor eax, esp
mov [esp+184h+var_4], eax
push offset aAlan ; "alan"
push 1 ; bInitialOwner
push 0 ; 1pMutexAttributes
call ds:CreateMutexA
test eax, eax
jz short loc_402A4B



We want to get the Mutex value for this malware : Writing basic python script

szusername= byte ptr -84n var 4= dword ptr -4									Basic script													
argc= dword ptr 4 argc= dword ptr 4 argv= dword ptr 8 envp= dword ptr 0Ch sub esp, 184h mov eax,security_cookie xor eax, esp											data = open('/tmp/malware.exe','rb').read()											
											pattern = 'CInvalidArgException'											
mov push push	[esp+184 offset a 1		;	eax "alar bInit		wner				m =	= da	ata	l[:da	ata.f	ind	l(pa	tter	'n)]	.rsp	olit	('\x0)0\x00')
push call	0 ds:Creat	eMuto		1pMut	exAt	trib	utes			<pre>print " Mutex\t:", m.rsplit('\x00\x00')[-3]</pre>											3]	
test jz	eax, eax short lo	c 402	2A4B									_										
-		_																				
0001E840) 4D 61	= 7A	69	6C 6					0 20				6D			74 (62				Mozilla/4.0 (Compatible;
0001E840 0001E858 0001E870) 4D 61 3 20 41	= 7A) 53	69 49	6C 6 45 2 2A 2	0 36	2E	30	3B 5	0 20 7 69 3 68	6E	64	6F 6F 65	6D 77 63		20	4E .		20	6C 35 6D	2E	31	MSIE 6.0;Windows NT 5.1
0001E858 0001E870 0001E888) 4D 61 3 20 41 9 29 00 3 74 21	= 7A 53 00	69 49 00	45 2	0 36 F 2A	2E 00	30 74	3B 5 65 6	7 69	6E 2E	64 64	6F	77	73 69	20 70	4E . 68 (54	20 72 74	35 6D 2E	2E 65 63	31 6E 6F	MSIE 6.0;Windows NT 5.1)*/*.tech.deciphermen t.netwww.microsoft.co
0001E858 0001E870 0001E888 0001E888	4D 61 3 20 41 3 29 00 3 74 21 0 6D 00	F 7A 53 0 00 E 6E 0 00	69 49 00 65 00	45 2 2A 2 74 0 61 6	0 36 F 2A 0 00 C 61	2E 00 00 6E	30 74 77 00	3B 5 65 6 77 7 00 0	7 69 3 68 7 2E 0 00	6E 2E 6D 43	64 64 69 49	6F 65 63 6E	77 63 72 76	73 69 6F 61	20 70 73 6C	4E 68 6F 69	54 65 66 64	20 72 74 41	35 6D 2E 72	2E 65 63 67	31 6E 6F 45	MSIE 6.0;Windows NT 5.1)*/*.tech.deciphermen t.netwww.microsoft.co malanCInvalidArgE
0001E858 0001E870 0001E888 0001E888 0001E888	4D 61 20 41 20 29 3 74 20 60 3 78	7A 53 00 60 6E 00 365	69 49 65 00 70	45 2 2A 2 74 0	0 36 F 2A 0 00 C 61 9 6F	2E 00 00 6E 6E	30 74 77 00	3B 5 65 6 77 7 00 0 00 0	7 69 3 68 7 2E	6E 2E 6D 43 43	64 64 69 49 4E	6F 65 63	77 63 72 76 74	73 69 6F 61 53	20 70 73 6C 75	4E 5 68 6F 69 70	54 65 66	20 72 74 41 6F	35 6D 2E 72 72	2E 65 63 67 74	31 6E 6F 45 65	MSIE 6.0;Windows NT 5.1)*/*.tech.deciphermen t.netwww.microsoft.co malanCInvalidArgE xceptionCNotSupporte
0001E858 0001E870 0001E888 0001E888	4D 61 20 41 20 29 20 74 21 6D 23 78 24 63	7A 53 00 6E 00 365 78	69 49 65 00 70 63	45 20 2A 20 74 00 61 60 74 60	0 36 F 2A 0 00 C 61 9 6F 0 74	2E 00 00 6E 6E 69	30 74 77 00 00 6F	3B 5 65 6 77 7 00 0 00 0 6E 0	7 69 3 68 7 2E 0 00 0 00	6E 2E 6D 43 43 43	64 64 69 49 4E 4D	6F 65 63 6E 6F 65	77 63 72 76 74 6D	73 69 6F 61 53 6F	20 70 73 6C 75 72	4E 5 68 (6F (69 (70 7 79 4	54 65 66 64 70	20 72 74 41 6F 78	35 6D 2E 72	2E 65 63 67 74 65	31 6E 6F 45 65 70	MSIE 6.0;Windows NT 5.1)*/*.tech.deciphermen t.netwww.microsoft.co malanCInvalidArgE



Too much hard-coded value

szusername= byte ptr -84n var 4= dword ptr -4									Basic script																
argc= dv argv= dv	argc= dword ptr -4 argc= dword ptr 4 argv= dword ptr 8 envp= dword ptr 0Ch											data = open('/tmp/malware.exe','rb').read()													
sub esp, 184h mov eax,security_cookie xor eax, esp											pattern = 'CInvalidArgException'														
mov push push	[esp+1 offset 1	84h-			''a]	lan''		wner					m =	= d	ata	a[:da	ata.	.fine	d(<mark>p</mark>	atte	ern))].rs	spli	t('\x	00\x00')
push call	0 ds:Cre	atel	lute	-	10	lute	xAt	trib	utes				print " Mutex\t:", m [-3]												
test jz	eax, e short		402	A4B		_	_				_														
0001E840		6F	7A	69	6C	6C	61	2F	34	2E	30	20	28	43	6F	6D	70	61	74	69	62	6C	65	ЗB	Mozilla/4.0 (Compatible;
0001E858				49				2E				69	6E	64	6F	77	73	20	4E	54		35			MSIE 6.0;Windows NT 5.1
0001E870			00	00		2F		00	74			68		64	65	63	69	70	68	65)*/*.tech.deciphermen
0001E888		2E		65	74	00	00	00	77		77			69	63	72	6F	73	6F	66		2E			t.netwww.microsoft.co
0001E8A0 0001E8B8			00 65	00 70	61 74	6C	61 6F	6E 6F				00 00		49 4F	6E 6F	76 74	61 53	6C 75	69 70	64 70		72 72			malanCInvalidArgE
0001E8D0			~~	70 63		70		0E 69		6E				4C 4D	65	74 6D	55 6F	72		70 45		63			<pre>xceptionCNotSupporte dExceptionCMemoryExcep</pre>
0001E8E8				6E	00	00	00	00		45				70	74	69	6F	6E		00		E8			tionA.
0001E900				00	FF	FF		00				00			41	00	00	00			00	00			l.A



We want to get the Mutex value for this malware :

```
Basic script
|SZUSERNAME= DUTE PTF -84N
var 4= dword ptr -4
argc= dword ptr 4
                                                      data = open('/tmp/malware.exe','rb').read()
arqv= dword ptr 8
envp= dword ptr 0Ch
                                                      pattern = 'CInvalidArgException'
        esp, 184h
sub
        eax, <u>security</u>cookie
mov
xor
        eax, esp
                                                      m = data[:data.find(pattern)].rsplit('\x00\x00')
mov
        [esp+184h+var 4], eax
        offset aAlan
push
                        ; "alan"
push
        1
                        : bInitialOwner
push
        0
                        ; 1pMutexAttributes
                                                       print " Mutex\t:", m[-3]
call
        ds:CreateMutexA
test
        eax, eax
iz
        short loc 402A4B
```

Disass script

```
disass = Disass32(path='/tmp/malware.exe', verbose=False)
```

```
if disass.go_to_next_call('CreateMutexA'):
```

address_mutex = disass.get_arguments(3, convention=STDCALL)

print " Mutex\t:", disass.get_string(address_mutex)



We want to jump in Thread

🔛 🖂 🖻	3	
1oc 40	1404:	
lea	eax, [ebp+Th	readId]
lea	ecx, [ebp+Pai	
push	eax	; 1pThreadId
push	0	; dwCreationFlags
push	ecx	; 1pParameter
push	offset Start	Address ; 1pStartAddress
push	0	; dwStackSize
push	0	; 1pThreadAttributes
mov	dword 405308	. 0
mov		er], offset dword 405308
call	ds:CreateThr	
mov	hThread, eax	

Disass script

```
disass = Disass32(path='/tmp/malware.exe', verbose=False)
```

if disass.go_to_next_call('CreateThread'):

```
startAddress = disass.get_arguments(3)
```

disass.set_virtual_position(startAddress)



We want to get the C&C :

	the second comparison of the second sec	
push	0	; dwContext
push	0	; dwFlags
push	3	; dwService
mov	edi, eax	
mov	eax, [esp+1043Ch	+1pszPassword]
push	eax	; 1pszPassword
push	ebp	; 1pszUserName
push	50h	; nServerPort
push	esi	; <pre>lpszServerName = "tech.decipherment.net</pre>
push	edi	; hInternet
mov	[esp+10450h+var	1040C], edi
call	ds:InternetConne	ctA
push	1F4h	; dwMilliseconds
mou	ohn oav	-

Disass script

if disass.go_to_next_call('InternetConnectA'):

print " CC1\t:", disass.get_string(disass.get_arguments(2))



DEFENDING WORLD SECURITY

Demo



© 2012 CASSIDIAN CYBERSECURITY - All rights reserved

Disass scripts vs. malware evolution

- Set a different C&C
- Bugs fixes in malware
- New features in malware
- Packing
- Encryption



Disass is available

Disass is available here (Alpha release):

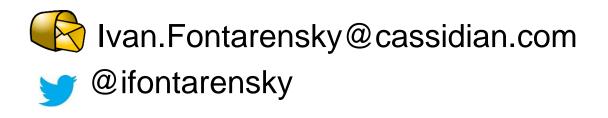
Disass support PE32 on x86

http://bitbucket.cassidiancybersecurity.com/disass



DEFENDING WORLD SECURITY

Questions ?





© 2012 CASSIDIAN CYBERSECURITY - All rights reserved

Distorm support Intel x86 8bit 16bit 32bit and 64bit

Elfesteem to manage ELF format

