Malware Analysis - DAY 3

Prep By Yohanes Syailendra

Today's Agenda

Fundamental Reverse EngineeringMalware Memory Analysis



Reverse Engineering

Reverse Engineering tools



- Radare2 is one of reverse engineering tool that can be used in linux environment(command Line) to analyze both windows and linux executables
- IDA Pro and OllyDbg is two reverse engineering tools used in windows environment to handle windows malware or applications

Immunity Break Down

🖏 Immunity Debugger - avtst2.exe - [CPU - main thread, module avtst2]		- 0 ×
C File View Debug Plugins ImmLib Options Window Help Jobs		- 8
🗁 🕉 🗟 🔣 📢 🗙 🕨 📕 🐓 🐉 🕌 🚽 🔸 lemtwhcPk	b z r s ? Immunity: Consulting Services Manager	
00401050 \$ 55 PUSH EBP		▲ Registers (FPU)
00401051 . 89E5 HOU EBP,ESP		EAX 771A3378 kernel32.BaseThreadInitThunk
00401053 . 6A FF PUSH -1		ECX 0000000
00401055 . 68 58114000 PUSH avtst2.00405000	SE handler installation	EDX 09401050 avtst2. <moduleentrypoint> EDX 7EEDEADA</moduleentrypoint>
0040105F . 64:FF35 00000 PUSH DWORD PTR FS:[0]		EDA / EP DEUBO
00401066 . 64:8925 00000 HOU DWORD PTR FS:[0],ESP		EBP 0818FF94 Registers willdow
0040106D . 83EC 0C SUB ESP,0C		
00401070 . 55 PUSH ESI		EDI 86868888 IS SEIL EXPLAINTORY
00401072 . 57 PUSH EDI	CPU window shows instructions	EIP 00401050 avtst2. <noduleentrypoint></noduleentrypoint>
00401073 . 8965 E8 HOU DWORD PTR SS:[EBP-18],ESP	ci e milaon silons instructions	C 0 ES 002B 32bit 0(FFFFFFF)
00401076 . 68 00000002 PUSH 2000000 80401078 F8 00000000 POLL sutst2 00401010		P 1 CS 0023 32bit 0(FFFFFFF)
00401080 . 59 POP ECX		A U SS 0928 32DIT U(FFFFFFF) 7 1 DS 0902 92bit 0(FFFFFFFF)
00401081 . A3 38434000 MOU DWORD PTR DS:[404338],EAX		S 6 FS 0053 32bit G(FFF)
00401086 . E8 E5020000 CALL avtst2.00401370		T Ø GS 002B 32bit 0(FFFFFFF)
00401088 . 8500 IEST EHX,EHX 00401080 75 00		
9949198F . 6A 91 PUSH 1		U 8 LASTEPP EKKUK_SKS_KEY_NUI_FUUND (80803087)
00401091 . E8 1A050000 CALL avtst2.00401580		EFL 08000246 (NO.NB,E,BE,MS,PE,GE,LE)
00401096 . 59 POP ECX		STØ empty g
00401097 . E9 96000000 JNP 40(St2.00401137 00401097 . E9 96000000 JNP 40(St2.00401137		ST1 empty g
004010A3 . E8 18050000 CALL avtst2.004015C0		sizempty y Sizemptu a
004010A8 . E8 D3050000 CALL avtst2.00401680		ST4 empty g
004010AD . E8 1E060000 CALL avtst2.004016D0		ST5 empty g
00401087 . E8 C40A0000 CALL avtst2.00401880		STC empty g
004010BC . BB 90304000 HOU EBX,avtst2.00403090		
004010C1 . 81FB 90304000 CMP EBX,avtst2.00403090		- FST 0900 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
004010C1 81FB 90304000 CMP EBX,avtst2.00403090 .00A01077 72 00 100 SUNDY autot2 00A0100A EBP-0018FF94		- FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT) FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1
094010C1 81FB 90304000 CHP EBX,autst2.00403090 		- FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT) FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1
094010C1 81FB 90304000 CHP EBX,autst2.00403090 4004010C7 72 00 HD SUNBT autst2.00403096 EBP=0018FF94		- FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT) FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1
094010C1 81FB 90304000 CHP EBX,autst2.00403090 004010C7 72 00 HD SU087 autst2.00403096 EBP-0018FF94 Address Hex dump ASCII	▲ 0018FF8C 771A338A ∎3-₩ RETU <u>RN to kernel32.7</u>	FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT) FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1
004010C1 .81FB 90304080 CHP EBX,autst2.00403090 .004010C7 72 00 .004010C7 .004010C7 .004010C7 </td <td>م 0018FF8C 771A330A ∎3-⊎ RETURN to kernel32.7 0018FF90 7EFDE000 مُنْ</td> <td>- FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 (GT) FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1 771A338A</td>	م 0018FF8C 771A330A ∎3-⊎ RETURN to kernel32.7 0018FF90 7EFDE000 مُنْ	- FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 (GT) FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1 771A338A
00461001 .01FB 90304000 CHP EBX,autst2.00403090 00461007 73 00 EBP-0010FF94 Address Hex dump 00464000 3C 40 00 00 00 00 00 00 00 00 00404000 00 00 00 00 40 41 00 0000A	 ▲ 0018FF8C 771A338A ■3-0 RETURN to kernel32.7 Ø018FF90 7EFDE000 .3ý[∞] Ø018FF94 @018FF04 Ûÿ, Ø049FF00 272707570 ₩4.051100 +	FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00401001 .01FB 96304000 CHP EBX,autst2.00403090 00501007 73 00 EBP-0010FF94 Address Hex dump 00404000 30 40 00 00 00 00 00 00 (0 00404000 00 00 00 00 00 00 (0 00404000 00 00 00 00 00 (0 00404000 00 00 00 00 00 00 (0 00404000 00 00 00 00 00 (0 00404000 00 00 00 00 00 00 (0 00404000 00 00 00 00 00 00 (0 00404000 00 00 00 00 00 00 (0 00404000 00 00 00 00 00 00 (0	 ▲ 0918FF8C 771A338A ■3-₩ RETURN to kernel32.7 ● 0618FF99 7EFDE 060 > 34[∞] ● 0618FF94 ● 0618FF94 ● 0618FF954 ○ 77979F72 ■ 0618FF95 ■ ■ 0618FF95	^{771A338A}
09401001 .81FB 96304080 CHP EBX,autst2.00403090 000401007 72 00 EBP=0018FF94 Address Hex dump 09404000 30 40 00 00 00 00 00 00 00 00 00 00 00 00	 ▲ 0018FF8C Ø018FF9C Ø018FF96 Ø018FF94 Ø018FF94 Ø018FF94 Ø018FF95 Ø018FF96 Ø18FF96 Ø18FF	- FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT) FGW 027F Prec NEAR,53 Mask 1 1 1 1 1 1 771A338A 79F72 Stack window shows
094010C1 81FB 98304080 CHP EBX, autst2.00403090 0004010C7 72.00 100 100 100 00401005 EBP-0018FF94 100 100 100 00011 00401005 Address Hex dunp ASCII 00404000 00	 ▲ 0018FF9C 771A338A ■3-₩ RETURN to kernel32.7 0018FF90 7EFDE000 .àý[∞] 0018FF94 r0018FF94 ôÿ1. 0018FF97 r018FF94 ôÿ1. 0018FF97 72 r■10 RETURN to ntdl1.7797 0018FF96 72EFDE000 .àý[∞] 0018FF97 72E671E ñæv 0018FF94 00009060 	^{79F72} Stack window shows contents of stack
094010C1 81FB 96304080 CHP EBX, autst2.00403090 0004010C7 72.00 100 SUBPT autst2.00403090 0004010C7 72.00 100 SUBPT autst2.00403090 004010C7 72.00 100 SUBPT autst2.00403090 004010C7 72.00 100 SUBPT autst2.00403090 004010C7 72.00 100 SUBPT autst2.0040300 00404008 00.00 00.00 00.00 SUBPT autst2.0040300 00404008 00.00 00.00 00.00 00.00 SUBPT autst2.0040300 00404008 00.00 00.00 00.00 00.00 SUBPT autst2.0040300 00404020 00.00 00.00 00.00 00.00 SUBPT autst2.00 00404020 05.00 00.00 00.00 00.00 SUBPT autst2.00 Shows met 00404020 00.00 00.00 00.00 00.00 Shows met	OB18FF8C 771A338A II3u RETURN to kernel32.7 0618FF94 76018F04 ôj'. 0618FF94 70018F04 ôj'. 0618FF94 70018F04 ôj'. 0618FF96 772979727 We RETURN to ntdll.7797 0618FF96 7720F712 ôj'' 0618FF96 7720F712 ôj'' 0618FF96 7720F712 ôj'' 0618FF96 7720F712 ôj'' 0618FF64 7760608 ôj'' 0618FF64 7760702 ôj'' 0618FF64 7760702 ôj'' 0618FF64 7760702 ôj'' 0618FF64 060908000 0618FF68 ój'' ój''	FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0
B04641001 B1FB 98304888 CHP EBX, autst2.08483998 B04641007 73 BD EBP-8018FF94 HM Stant autst2.08483998 Address Hex dump ASCII 80494089 36 C49 59 69 80 50 68 69 90494089 30 0 60 69 48 41 60 69 66 C1 90494018 80 69 69 68 68 69 66 C1 90494028 50 69 69 69 69 69 69 69 69 69 90494028 50 69 69 69 69 69 69 50 50 69 90494028 50 89 69 69 69 69 69 50 50 69 69 69 69 90494028 50 89 69 69 69 69 69 50 50 69 69 69 69 69 90494028 50 89 69 69 69 69 69 50 50 60 69 69 69 69 90494028 50 89 69 69 69 69 69 50 69 69 90494028 50 89 69 69 69 69 69 50 69 90494028 50 89 69 69 69 69 69 50 69 90494028 50 89 69 69 69 69 69 50 69 90494028 69 89 69 69 69 69 50 69 90494028 69 89 69 69 69 50 69 90494028 69 89 69 69 69 50 69 90494028 69 89 69	▲ 0918FF8C 771A338A ■30 RETURN to kernel32.7 0918FF99 7EFDE060 .3ý [∞] 9018FF94 9018FF94 09172 rHu RETURN to ntd11.7797 0018FF95 7EFDE080 .3ý [∞] 0018FF40 0018FF40 0018FF40 0018FF40 0018FF40 0018FF40 00800000 0018FF40 00800000 0018FF40 00800000 0018FF40 00800000 0018FF40 00800000 0018FF40 00800000	<pre> FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>
004041001 0176 96304000 CHP EBX, Jut 5t2 004083090 00501007 73 00 100 51007 1121 004083090 00501007 73 00 100 51007 1121 004083090 00404500 30 40 00 60		^{771A338A} ^{79F72} Stack window shows contents of stack
094010C1 01FB 96304000 CHP EBX, autst2.00403090 0001010C2 73 00 EBP=0010FF94 Address Hex dump 09404000 3C 40 00 00 00 00 00 00 00 (0 09404000 3C 40 00 00 00 00 00 (0 09404000 3C 40 00 00 00 00 00 (0 09404000 3C 40 00 00 00 00 00 (0 09404000 3C 40 00 00 00 00 00 (0 09404000 3C 40 00 00 00 00 00 (0 09404000 3C 40 00 00 00 00 00 (0 09404000 5C 41 00 00 12 (1.4) 09404030 5C 41 00 00 12 (1.4) 09404030 00 00 00 00 00 00 00 00 00 00 00 00 00	▲ 0018FF8C 771A338A ■3-₩ RETURN to kernel32.7 0018FF99 7EFDE 000 .3ý [∞] 0018FF94 0018FF94 0ij [±] . 0018FF95 7779772 r=111 RETURN to ntd11.7797 0018FF96 77E6F018 .3ý [∞] 0018FF64 00090000 0018FF64 00090000 0018FF64 0018FF68 00090000 0018FF88 00090000 0018FF88 00090000 0018FF88 00090000 0018FF88 00090000 0018FF88 00090000	FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0
09401001 81FB 98394090 CHP EBX, autst2.00403990 000401007 72 00 EBP-0018FF94 Address Hex dump 09404000 30 00 00 00 00 00 00 00 00 00 00 00 00 0		FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0
09401001 81FB 98304080 CHP EBX, autst2.0040399 000401007 72.00 100 SHDD1 autst2.00403996 000404008 32.40 00 00 00 00 000404008 32.40 00 00 00 00 00 00040408 00 00 00 00 00 00 00 00040408 00 00 00 00 00 00 00 00 000404032 00	O018FF8C 771A338A IS-u RETURN to kernel32.7 0018FF90 7EFDE000 .àý~ 0018FF94 70018FF04 Öÿ1. 0018FF96 772979727 RETURN to ntdl1.7797 0018FF96 77209712 Numeration 0018FF96 77209712 Numeration 0018FF96 72667116 New 0018FF64 7726972 Numeration 0018FF64 90909000 0018FF64 5000 .àý~ 0018FF64 00909000 0018FF84 00909000 0018FF84 00909000 0018FF84 00909000 0018FF84 00909000 0018FF84 00909000 0018FF86 08090000 0018FF86 08090000 0018FF86 08090000 0018FF86 08090000 0018FF86 08090000 0018FF86 <td><pre> FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre></td>	<pre> FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>
004041061 01FB 98304000 CHP EBX, autst2.00483090 005041061 01FB 98304000 CHP EBX, autst2.00483090 005040400 0150 0160 0160 0160 0160 0160 0160 0010 00404000 0180 00	A 0018FF8C 771A338A ■3v RETURN to kernel32.7 0018FF90 7EFDE000 .àý [∞] 0018FF94 9018FF94 3018FF94 3018FF94 3018FF94 0018FF94 9018FF94 301869 0018FF40 9018FF40 3018969 0018FF40 9019090 0018FF40 9019090 0018FF41 9090909 0018FF42 9090909 0018FF84 9090909 0018FF84 9090909 0018FF84 9090909 0018FF84 9090909 0018FF84 9090909 0018FF84 9090909 0018FF85 9090909 0018FF86 9090909 0018FF86 9018F640 ÿ1. 0018FF62 9018F640 ÿ1. 0018FF63 77927157 501₩ SE handler	<pre> F5T 8000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>
0040410C1 01FB 96304000 CHP EBX, Jutst2.00403090 005010C7 73 00 EBP-0010FF94 Address Hex dump 00404000 3C 40 00 00 00 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 00 00 00 00 00404000 00 00 00 00 00 00 00 00 00404000 00 00 00 00 00 00 00 00 00 00404000 3C 41 00 00 00 00 00 00 00404000 32 41 00 00 68 41 00 00 1\$A 00404040 32 41 00 00 82 41 00 00 1\$A 00404040 32 41 00 00 82 41 00 00 1\$A 00404040 32 41 00 00 82 41 00 00 1\$A 00404040 32 41 00 00 68 41 00 00 1\$A 00404040 22 41 00 00 68 41 00 00 1\$A 00404040 22 40 00 00 A1 40 00 00 1\$A 00404040 22 40 00 00 A1 40 00 0 1\$A 00404040 22 40 00 00 A1 40 00 0 A1 00 00 A1 00404040 22 40 00 00 A1 40 00 0 A1 00 00 A1 00404040 0 C2 41 00 00 FA 11 00 00 A1 004040407 00 C2 40 00 00 A1 42 00 00 A1 004040407 00 C4 20 00 00 A 42 00 00 A1		<pre> F5T 8000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>
004/01/01 01/01 00		<pre> F5T 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>
094010C1 81FB 98304000 CHP EBX, Jutst2.00403090 0004010C2 72 00 100 SH0BT Jutst2.00403090 EBP=0018FF94 100 SH0BT Jutst2.0040306 00401065 Address Hex dump ASCII 09404080 3C 40 00 00 00 00 00 00 00 00 ASCII 09404080 3C 40 00 00 00 40 41 00 00	A 0018FF8C 771A338A ■3-₩ RETURN to kernel32.7 0018FF99 7EFDE009 .3ý~ 0018FF94 0018FF9 0018F94	79F72 Stack window shows contents of stack
0040410c1 ett FB 98304090 ctr FB 98304090 Colspan="2">Colspan="2" Colspan="2">Colspan="2" Colspan="2" Colspan="2" Colspan="2" Colspan="2" Colspan="2">Colspan="2" Colspan="2" Colspan="2" <thcolspan="2"< th=""> <thcolspan="2"< th=""></thcolspan="2"<></thcolspan="2"<>	Constrained Constrai	771A338A 79F72 Stack window shows contents of stack 79F45 from ntd11.77979F48
004041061 015E 98304000 CHP EBX, Jutst2.00483090 005041061 015E 94 00404060 02 40 00 5000 monocol 000 5000 monocol 00404060 02 40 00 60 00 00 00 00 00 00 00 00 00 5000 monocol 000 00 00 00 00 00 00 000 00 00 00 00 00 000 00 00 00 00 00 000 00 00 00 00 000 00 00 00 00 00 000 00 00 00 000 00 000 00 00 000 00 00 000 00 00 000 00 000 00 000 00 000 00 000 00 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00	A 0018FF8C 771A338A ■3-u RETURN to kernel32.7 0018FF90 7EFDE080 .30 [∞] 0018FF94 0018FF04 0ÿ1. 0018FF94 0018FF04 0ÿ1. 0018FF40 77E6F118 ñæu 0018FF40 00809069 0018FF40 00809069 0018FF40 00809069 0018FF42 00809069 0018FF82 00809069 0018FF62 008080909 0018FF64 018FF62 008090909 0018FF64 011.7797 0018FF60 0040169 0018FF62 0040169 0018FF64 00809090 0018FF64 011.7797 0018FF69 7799745 EIW RETURN to ntd11.7797 0018FF69 7797945 EIW eIURN to ntd11.7797 0018FF69 7E6060 3y [∞]	771A338A 79F72 Stack window shows contents of stack 79F45 from ntd11.77979F48
0040410C1 01FB 96304000 CHP EBX, autst2.00403090 0050410C7 73 00 100 Stind 1 autst2 00403090 0050404007 73 00 100 Stind 1 autst2 00403090 00404000 30 400 00	• 0018FF8C 771A338A 13u RETURN to kernel32.7 • 0018FF90 7EFDE060 .30° • 0018FF94 • 0018FF04 001°. • 0018FF94 • 0018FF04 001°. • 0018FF94 • 0018FF04 00°. • 0018FF94 • 0018FF04 00°. • 0018FF94 • 0000000 • 0018FF94 • 0000000 • 0018FF64 • 0018FF64 • 0018F64 • 0018FF64 • 0018F60 • 0018F64 • 0018FF64 • 0018F60 • 0018F64 • 0018FF60 • 0018F66 • 0018F64 • 0018F670 • 0018F650 • 0018F65 • 0018F760 • 0018650 • 0018F66	771A338A 79F45 from ntd11.77979F48 790int>
0040410c1 .01FB 96304000 CHP EBX.autst2.00403090 0050410c7 73 00 EBP-0013FF94 00404000 3C 40 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 00 00 00 00 00 00404000 3C 40 00 00 00 08 40 00 00 00 00404000 3C 40 00 00 00 08 40 00 00 00 00404010 80 40 00 00 08 40 00 00 00 00 00404020 5C 41 00 00 1C 41 00 00 1A.A. 00404030 00 00 00 00 00 00 00 00 00 00 00 00	Coll8FF8C 771A338A ■3-0 RETURN to kernel32.7 O018FF8C 77507972 r=W RETURN to kernel32.7 O018FF94 O018F94 O018F	77 FST 0000 Cond 0 0 0 0 0 Err 0 0 0 0 0 0 0 0 0 0 0 0 0
B04010C1 81FB 96304080 CHP EBX, Jutst2.00403090 0001010C2 72 00 100 SH0FT Jutst2.00403004 EBP-0018FF94 100 SH0FT Jutst2.00403004 004034000 004034000 3C 40 00 00 00 00 00 00 (0 004034000 3C 40 00 00 00 00 (0 004034000 004034000 3C 40 00 00 00 00 00 (0 004034000 3C 40 00 00 00 00 00 (0 004034000 3C 40 00 00 00 00 (0 004034000 5C 410 00 00 00 00 00 00 00 00 004034020 5C 41 00 00 12 41 40 00 1 004034020 00 00 00 00 00 00 00 00 00 00 004034020 5C 41 00 00 00 00 00 00 00 00 00 004034020 00 00 00 00 00 00 00 00 00 00 004034020 00 00 00 00 00 00 00 00 004034020 00 00 00 00 00 00 00 00 00 004034020 00 01 00 00 00 41 00 00 1 004044020 02 41 00 00 00 4 004044020 02 41 00 00 7 004044050 00 41 00 00 92 41 00 00 1 004044050 00 41 00 00 92 41 00 00 4 000 404070 0C 42 00 00 11 42 00 00 40 004044050 22 41 20 00 00 12 41 00 00 40 004044050 22 41 20 00 00 12 41 00 00 40 004044050 50 42 42 00 00 12 41 40 00 00 40 004044050 50 42 42 00 00 12 41 40 00 00 40 004044050 50 42 42 00 00 12 41 40 00 00 40 004044050 50 42 42 00 00 12 41 40 00 00 40 004044050 50 42 42 00 00 12 41 40 00 00 40 004044050 50 42 42 00 00 12 42 40 00 00 00 00 00 00 00 00 00 00 00 00	> 0018FF8C 771A338A US-u RETURN to kernel32.7 0018FF90 7EFDE000 .àj~ 0018FF94 7018FD4 001. 0018FF94 7720F727 RETURN to ntdll.7797 0018FF96 7FDE000 .àj~ 0018FF96 7FDE000 .àj~ 0018FF96 7FDE000 .àj~ 0018FF96 7FDE000 .àj~ 0018FF96 06909060 0018FF86 06909060 0018FF60 06809	77163380 77163380 79772 Stack window shows contents of stack 79745 from ntd11.77979F48
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	Constrained Constrai	771A338A 79F72 Stack window shows contents of stack 79F45 from ntd11.77979F48 Point>
0040410C1 1FB 98304000 CHP EBX, autst2.00483090 00504000 73 00 100 Stn01 autst2 0040400 EBP-0018FF94 00 60 00 60	Constrained in the interval of the inter	FST 8000 Cond b b b c Err b b b b b b b b b b b b b b b b b b
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		771A338A 79F72 Stack window shows contents of stack 79F45 from ntd11.77979F48 Point> Point>

Immunity Breakdown 2

🖧 Immunity Debugger - avtst2.exe - [CPU - main thread, module avtst2]	-							
C File View Debug Plugins ImmLib Options Window Help Jobs								
🗁 🐝 🗏 🔣 🐳 × ▶ ▶ ╄ 🛊 🔰 🖊 → 🕴 lemtwh c Pkbzrs? 🛛 Immunity: Consulting Services Manager								
00401050 \$ 55 PUSH EBP								
00401051 . 89E5 MOV EBP,ESP								
00401053 . 6A FF PUSH -1 Keterences								
00401055 . 68 0C304000 PUSH avtst2.0440.00C								
89491971 . 64:8925 8888811031 DWORD PTR 15:81 FSP								
0040106D . 83EC 0C SUB ESP, 0C								
00401070 . 53 PUSH EBX Software Breakpoints								
00401071 . 56 PUSH ESI								
00401072 . 57 PUSH EDI								
00401073 . 8965 E8 MOU DWORD PTR ST:[EPP-18], esp								
0040107B . E8 900.0000 LALL 40CSL2.00401010								
AA41088 . 37								
00401086 . E8 E5020000 CALL autst2.0040 870 Show CPU window								
0040108B . 85C0 TEST EAX,EAX								
0040108D . 75 0D JNZ SHORT avtst2.1040109C								
0040108F . 6A 01 PUSH 1								
00401091 . E8 1A050000 CALL avtst2.00401980 Snow Handles								
60401068 E8 03050000 Gall autst2.00401688 Show Threads								
004010AD . E8 1E060000 CALL autst2.004016D								
004010B2 . E8 290A00000 CALL avtst2.00401AE								
004010B7 . E8 C40A0000 CALL autst2.00401B80 Show Memory Map								
004010BC . BB 90304000 MOV EBX,avtst2.00403090								
004010C1 . 81FB 90304000 CMP EBX,avtst2.00403190								
ERPERING Show Loaded modules								
Show log								

Intro to Assembly Common Instructions:

Registers:

are sections of memory that can be quickly accessed on the CPU die EIP (Instruction pointer) and ESP (Stack pointer) are used for pointing to locations in memory while the majority of the other registers are used for general purposes

There is also a flags register that can state various information about the CPU

Stack:

A section of memory that contains currently used data

Intro to Assembly Registers:

EAX	Primary Accumulator - stores function return values
ECX	Count Register - Counter for string and loop operations
EDX	Data Register - I/O pointer
EBX	Base Register - Base pointer to the data section
ESP	Stack pointer
EBP	Base Pointer - Stack frame base pointer
ESI	Source Index- Source pointer for string operations
EDI	Destination Index - Destination pointer for string operations

EIP Instruction Pointer - Pointer to next instruction to execute

Intro to Assembly

Common Instructions:

NOP - No Operation

PUSH - Moves a word/Dword/Qword or register (not EIP) onto the stack

POP - Removes a Dword off the stack and puts it in a register

CALL - Transfers control to a different function in a way that control can be returned back

(A call can take place using an absolute or relative address)

RET - Used to return from a function

 ${\sf MOV}$ - Can move a register to a memory / memory to a register, an immediate to register / immediate to memory

LEA - copy the result of one operand (register/memory/address/constant) to another

CMP - Compares two operands

JMP - Moves control to absolute or relative address

The following conditional jumps perform a JMP based on the condition of the previous CMP

JE - When equal | JNE - When not equal || JZ - When zero | JNZ - When not zero

JG - When greater than | JGE - When greater than or equal || JL - When less than

JLE - When less than or equal to

Install GCC

Test if there's gcc installation

\$#locate glibc =>

It should be :"/usr/share/man/man7/glibc.7.gz"

 \triangleright \$#gcc \rightarrow should be:

"gcc: fatal error: no input files" => already installed

► To install GCC :

\$#apt-get install gcc

Create Hello file.c

\$ nano file.c

#include <stdio.h>
int main()
{
 printf("Haloo");
 return 0;
 }

Open at edb

- Search for Helloo string and replace with another string
- Edit the string to another

Search for the Flag

- R2 [filename]
- ► Type 'aa' → to start analyze all
- ► Type 'pdf@main' → to find the int main function
- Find the flag:
 - http://libra.syailendra.my.id/download/malware-analysis/test
 - http://libra.syailendra.my.id/download/malware-analysis/wow2
 - http://libra.syailendra.my.id/download/malware-analysis/test2
 - http://libra.syailendra.my.id/download/malware-analysis/test3
 - http://libra.syailendra.my.id/download/malware-analysis/test4
 - http://libra.syailendra.my.id/download/malware-analysis/test5

Search For the Flag

- Open Windows
- Install the tools:
 - Immunity Debugger
 - Die it Easy 0.95
 - IDAPro
- Check the Password Flag
 - http://libra.syailendra.my.id/download/malware-analysis/App1.exe
 - http://libra.syailendra.my.id/download/malware-analysis/App2.exe
 - http://libra.syailendra.my.id/download/malware-analysis/App4.exe
 - http://libra.syailendra.my.id/download/malware-analysis/App5.exe
 - http://libra.syailendra.my.id/download/malware-analysis/App6.exe
 - http://libra.syailendra.my.id/download/malware-analysis/App7.exe

Memory Analysis

Volatility

jnieto@behindthefirewalls:/home/volatility-2.1\$ python vol.py -f zeus.vmem pstree											
Volatile Systems Volatility Framework 2.1											
Name	Pid	PPid	Thds	Hnds	Time						
0x810b1660:System	4	0	58	379	1970-01-01	00:00:00					
. 0xff2ab020:smss.exe	544	4	3	21	2010-08-11	06:06:21					
0xff1ec978:winlogon.exe	632	544	24	536	2010-08-11	06:06:23					
<pre> 0xff255020:lsass.exe</pre>	688	632	21	405	2010-08-11	06:06:24					
<pre> 0xff247020:services.exe</pre>	676	632	16	288	2010-08-11	06:06:24					
<pre> 0xff1b8b28:vmtoolsd.exe</pre>	1668	676	5	225	2010-08-11	06:06:35					
0xff224020:cmd.exe	124	1668	0		2010-08-15	19:17:55					
<pre> 0x80ff88d8:svchost.exe</pre>	856	676	29	336	2010-08-11	06:06:24					
<pre> 0xff1d7da0:spoolsv.exe</pre>	1432	676	14	145	2010-08-11	06:06:26					
<pre> 0x80fbf910:svchost.exe</pre>	1028	676	88	1424	2010-08-11	06:06:24					
0x80f60da0:wuauclt.exe	1732	1028	7	189	2010-08-11	06:07:44					
0x80f94588:wuauclt.exe	468	1028	4	142	2010-08-11	06:09:37					
<pre> 0xff364310:wscntfy.exe</pre>	888	1028	1	40	2010-08-11	06:06:49					
<pre> 0xff217560:svchost.exe</pre>	936	676	11	288	2010-08-11	06:06:24					
<pre> 0xff143b28:TPAutoConnSvc.e</pre>	1968	676	5	106	2010-08-11	06:06:39					
<pre> 0xff38b5f8:TPAutoConnect.e</pre>	1084	1968	1	68	2010-08-11	06:06:52					
<pre> 0xff22d558:svchost.exe</pre>	1088	676	7	93	2010-08-11	06:06:25					
<pre> 0xff218230:vmacthlp.exe</pre>	844	676	1	37	2010-08-11	06:06:24					
<pre> 0xff25a7e0:alg.exe</pre>	216	676	8	120	2010-08-11	06:06:39					
<pre> 0xff203b80:svchost.exe</pre>	1148	676	15	217	2010-08-11	06:06:26					
<pre> 0xff1fdc88:VMUpgradeHelper</pre>	1788	676	5	112	2010-08-11	06:06:38					
0xff1ecda0:csrss.exe	608	544	10	410	2010-08-11	06:06:23					
0xff3865d0:explorer.exe	1724	1708	13	326	2010-08-11	06:09:29					
. 0xff374980:VMwareUser.exe	452	1724	8	207	2010-08-11	06:09:32					
. 0xff3667e8:VMwareTray.exe	432	1724	1	60	2010-08-11	06:09:31					

Download the images

- http://libra.syailendra.my.id/download/malware-analysis/cridex.zip
- http://libra.syailendra.my.id/download/malware-analysis/zaptftis.rar

Volatility

- ./vol.py imageinfo -f <Destination of the memory Dump>
- ./vol.py -profile=WinXPSP2x86 pslist -f <Destination of the memory Dump> show all running process
- ./vol.py -profile=WinXPSP2x86 kdbgscan -f <Destination of the memory Dump> → show kernel debugger block (show hidden process)
- ./vol.py -profile=WinXPSP2x86 kpcrscan -f <Destination of the memory Dump>
 show processor specific data
- ./vol.py -profile=WinXPSP2x86 dlllist-f <Destination of the memory Dump> show all running dll
- ./vol.py -profile=WinXPSP2x86 dlldump -D <Destination Directory> -f <memory image location> → Dump all DLL into folder

Volatility

- ./vol.py -profile=WinXPSP2x86 psscan-D <Destination Directory> -f <memory image location> → scan all process
- ./vol.py -profile=WinXPSP2x86 -f <memory image location> > Show all process in a tree
- ./vol.py -profile=WinXPSP2x86 connection -f <memory image location>
 Show all running connection./vol.py -profile=WinXPSP2x86 sockets -f <memory image location> > show all open sockets (ports)
- ./vol.py -profile=WinXPSP2x86 hivescan -f <memory image location> → search for any injected process
- ./vol.py -profile=WinXPSP2x86 hivelist -f <memory image location> → search for any injected process on virtual memory
- ./vol.py -profile=WinXPSP2x86 svcscan -f <memory image location> → show all services on memory

Thank You