SafeSEH+SEHOP all-at-once bypass exploitation method principles

x90c Jan 8. 2012

[Table of Contents]

- 1. Which windows are SEHOP contained?
- 2. The explotation method principles
- 3. Limitations
- 4. Other mitigations ?
- 5. References



***some** windows system module's _except_handler3 hasn't any checking of user-defined Handler address(*loc1). Then we can use it for calling our shellcode indirectly.

1. Which windows are SEHOP contained?

- Windows Vista sp1 (Optional)
- Windows 7 (Optional)
- Windows Server 2008 (Default enabled)
- Windows Server 2008 R2 (Default enabled) [1]

2. The exploitation method principles

There are two principles of SEHOP all-at-once bypass method.

- 1. Targeted(or victim) stackoverrun vulnerable application built by VC++ 6.0 and some other versions(non-tested).
- 2. The vulnerable application running on a box which SEHOP mitigation enabled windows environment.

👰 Pid 2824 - WinDbg:6,11,0001,404	X86				_ 8 ×
File Edit View Debug Window Help					
2 3 10 11 12 25 10 (0 0 0 10 0 2 0 10 10 1					
Memory 🔤	Disassembly			E.	Registers 🙀 🖬
Virtual: Previous	Uffset: @Sscopeip			Previous Next	Customize
Display format: Byte	No prior disassembly possib 0012e4d8 cc	ole int	3	Platform: MS Win Vista sp1 Ultimate ko	Reg Value
Next	0012e4d9 cc	int	3	Flationne wie vier vista spr oltinate ko	gs 0
0012e4d8 cc cc cc cc 🔺	0012e4da CC 0012e4db CC	int	3		fs 3b
0012e4dc 41 41 41 41 AAAA	0012e4dc 41	inc	ecx	Onfo OF LLO OF LOOD all at an on the second through	es 23
0012e4e4 41 41 41 41 AAAA	0012e4dd 41 0012e4de 41	inc	ecx	SafeSEH+SEHOP all-at-once bypass Attack.	us 23
0012e4e8 41 41 41 41 AAAA 0012e4ec 41 41 41 41 AAAA	0012e4df 41	inc	ecx		esi 9
0012e4f0 41 41 41 41 AAAA	0012e4e0 41 0012e4e1 41	inc	ecx		ebx 12e5dc
0012e4f8 41 41 41 41 41 AAAA	0012e4e2 41	inc	ecx	0xcc trap shellcode executed!	edx 76e19bad
0012e4fc 41 41 41 41 ÅÅÅÅ	0012e4e3 41	inc	ecx		ecx 1b
0012e500 41 41 41 41 AAAA	0012e4e5 41	inc	ecx		eax 12da3U
0012e508 41 41 41 41 AAAA	0012e4e7 41	inc	ecx		ein 12e4d8
0012e510 41 41 41 41 AAAA	0012e4e8 41	inc	ecx		cs 1b
0012e514 41 41 41 41 AAAA	0012e4ea 41	inc	ecx		ef1 200206
0012e51c 41 41 41 41 AAAA	0012e4eb 41 0012e4ec 41	inc	ecx		esp 12da14
0012e520 41 41 41 41 41 44	0012e4ed 41	inc	ecx		ss 23
0012e528 41 41 41 41 AAAA	J 0012e4ee 41 0012e4ef 41	inc	ecx		dr0 0
0012e52c 41 41 41 41 AAAA	0012e4f0 41	inc	ecx		dr2 0
	0012e4f1 41	inc	ecx		
Command Data Calls					
0 e 7278bb44 0001 (0001) 0:**** MSVBVM60!CreateIExprSrvObj+0xa39					
Breakpoint 0 hit					
e_{a} =27012430 ebx-60124318 ebx-60124318 ebx-60104019 ebx-6010400 ebx-60124318 extends frame if hot in any known mouther following frames may be wrong. e_{a} =27278b44 esp-00124518 ebx-60124318 ext2e4d8					
cs=001b ss=0023 ds=0023 ds=0023 fs=003b gs=0000 efl=(01 0012e5f8 02dfec3e 02e70210 00000040 00000026 MSVBVM60(CreateIExprSrvDbj+Vax3d MSVBVM60(CreateIExprSrvDbj+Vax3d MSVBVM60(CreateIExprSrvDbj+Vax5t Max3d MSVBVM60(CreateIExprSrvDbj+Vax5t MAx5d MSVBVM60(CreateIExprSrvDbj+Vax5t MAx5d MSVBVM60(CreateIExprSrvDbj+Vax5t MAx5d MSVBVM60(CreateIExprSrvDbj+Vax5t MSVBVM60(CreateIEx					
1278bb44 ff548104 call dword ptr [edi+ecx*4+4] ds:0023:0012e 02 0012e67c 024381cd 003489e4 00120001 0342460 AdjmasEngidjwsZerjelimeEnhancedterter.					
0:000 t = 04 0012e694 10004528 00000000 003489e4 0000001 AdjMasEngld;PlayListLoad+0x37 2 = 00000000 00349e4 0000001 AdjMasEngld;PlayListLoad+0x37 2 = 00000000 00349e4 00000001 AdjMasEngld;PlayListLoad+0x37 2 = 00000000 00349e4 00000001 AdjMasEngld;PlayListLoad+0x37 2 = 0000000000 00349e4 00000001 AdjMasEngld;PlayListLoad+0x37 2 = 0000000000000000000000000000000000					
eip=0012e4d8 esp=0012da14 ebp=0012e5ec iop1=0 nv up ei pl nz r 06 0012e744 100375f 1004f274 0012e6d0 0000000 ubp3dj D11CanUnloadNow+Ux8d50					
cs=001b ss=0023 ds=0023 es 0012e4d8 cc int	s=0023 is=0036 gs=0000	et	1=1 0	17 0012e7cc 72709738 00000001 000000d 726bae58 amp3dj!DllCan 18 0012e808 727790f8 0236461c 0000000d 726bae58 MSVBVM601BAST	UnloadNow+0x82f0
	-		- O	09 0012e85c 727a7946 0236461c 0000000d 726bae58 MSVBVM60!TipI	nvokeMethod+0x3b6
				Ja UU12e8cU 727a8b8e U236461c UU000000d 00000001 MSVBVM60!_vba Nb 0012ea50 004c1f04 0035f240 0012ead0 0012eba0 MSVRVM60! vba	VarSub+Ux15c LateIdCall+0x1a
Ln 0, Cel 0 Sys 0: <lecal> Proc 000:508 Thrd 000:da0 ASM 🧇 A 漢 😢 🗧</lecal>					
灯 시작 📄 🧝 🏉 🛛 🗔 Pid 282	24 - WinDb 🎚 audiotran	🄄 🚱 Audio	itran		« 📵 民 🕼) 오전 1:09

3. Limitations

- 1. The method tested was with __except_handler3, not with __except_handler4.
- The method tested was with not related with __EH4_CallFilterFunction of scopetable struct. It was using the SEH chain only.
- The method was works by using __except_handler3-like copied routine around DLL's mapping area. (tested on Vista sp1 with SEHOP enabled manually.)
- __except_handler4 also have the routine to call user-defined handler. then if you can, also abusing this. as __EH4_CallFilterFunction of ScopeTable abusing. after GS security cookie, EH security cookie checking (XORed). (some good references are exists on googling.)

4 Other mitigations?

• Stack ASLR In Vista sp1?

"Microsoft's Windows Vista (released January 2007), Windows Server 2008, Windows 7, and Windows Server 2008 R2 have ASLR enabled by default, although only for those executables and dynamic link libraries specifically linked to be ASLR-enabled.[7] This did not include Internet Explorer 7 on Windows Vista prior to Service Pack 1; ASLR and Data Execution Prevention (DEP) are both disabled for application compatibility purposes." [3] As you can see, Vista I tested only enabled for Executables and DLLs. No Stack ASLR. and DEP is exists and can be applied to mapping areas. Vista, Windows Server 2008, Windows 7 is the SEHOP systems.

But if you want to enable stack ASLR on Vista. then it's also able to do.

"When executing a program whose image has been marked for ASLR, the memory layout of the process is further randomized *by placing the thread stack and the process heaps randomly.*" [4]

As you may know, Process Heaps also can be ramdomized.

• DEP?

The method was not considered about DEP(Data Execution Prevention).

5. References

- [1] http://support.microsoft.com/kb/956607
- [2] http://support.microsoft.com/kb/956607/en
- [3] http://en.wikipedia.org/wiki/Address_space_layout_randomization
- [4] http://www.symantec.com/avcenter/reference/Address_Space_Layout_Randomization.pdf
- [5] AudioTran 1.4.2.4 SafeSEH+SEHOP Exploit. http://www.exploit-db.com/exploits/15184