SEH all-at-once attack

a New technique to bypass SafeSEH+SEHOP protections

Table of Contents

- 1. What is SEH ?
- 2. E_R struct
- 3. E_R (windbg)
- 4. classical SEH overwrite
- 5. SafeSEH, SEHOP
- 6. History of SEH Protections bypass
- 7. SEH All-at-Once attack method
- 8. Case by case exploit
- 9. End.

1. What is SEH?

- MS Windows system's StackShield.
- SEH(Structured Exception Handling)
- fs:[0] saves nt!_TEB and first member field is nt!_TIB (not a pointer). nt!_TIB's first field is latest installed SEH E_R struct address.

2. E_R struct(Exception_Registration)

- | *Next | <- [ebp-10h]
- |*Exception Handler | <- [ebp-Ch]
- |an address of Image| <- [ebp-8]
- | index to User-defined exception handler | <- [ebp-4]
- | SFP |
- | RET |
- | ARGS |

- ***Next:** 4-byte next E_R struct pointer. Next E_R's handler is older than current E_R. (think about queue)
- ***Exception Handler:** 4-byte Exception Handler function address.
- ***an address of Image:** It used for calculating user-defined exception handler's address with next field(index) on _except_handler3 or _except_handler4 compiler generated exception handler.
- **Index:** it is the count of '__try{ … __try{' coding from count 0FFFFFFFFh.
- ex) __try{ } __except(…){} then this count is zero(0)
- __try{ … __try{ then this count is 1.

3. E_R (windbg)

```
0:000> dt _EXCEPTION_REGISTRATION_RECORD
ntdll!_EXCEPTION_REGISTRATION_RECORD
   +0x000 Next             : Ptr32 _EXCEPTION_REGISTRATION_RECORD
   +0x004 Handler          : Ptr32    _EXCEPTION_DISPOSITION
```

- Above two fields are E_R struct’s member.

- Note. I says E_R struct on this presentation is
-         (Next | Handler | [ebp-8] | [ebp-4]) == 16 bytes.
- Note II.
- Under applications built by **‘Debug’ mode**, Handler gots kernel32!_except_handler3 address and [ebp-8], [ebp-4] will used to calculate user-defined exception handler.
- Under **‘Release’ mode**, Handler gots the address of compiler generated Image!_except_handler3 and also works as ‘Debug’ mode.

4. classical SEH overwrite

- | vuln buffer | E_R struct |SFP|RET|SHELLCODE|
- |_______________^

- Step 1. overflow the stack vuln buffer to E_R struct.
  (E_R.Next = '₩xeb₩x06₩x90₩x90')
  (E_R.Handler = address of 'pop; pop; ret;' sequence bytecode.)
  Looking for the address of 'pop esi; pop esi; ret
  ('₩x5e₩x5e₩xc3') or 'add esp, 8;ret' to ROP(Return-Oriented-Programming).
  And overwrite the address onto E_R.Handler.

- .

Cont.

When Exception Ocurred:

- E_R.handler will called when after exception occurred.
- (Exception Dispatcher -> Exception Handler caller routines -> E_R.handler called!)
- when before calling E_R.handler, [esp+8] gots the address &E_R struct.
- ‘pop pop ret’ pops 8 bytes and returned to &E_R. then the overflowed short jmp will executed. ‘₩xeb₩x06 (short jmp $+6)’. This jmp short jumping into shellcode

5. SafeSEH, SEHOP

- SafeSEH protection:
- - E_R.handler must not pointers image area and windows DLL module address ranges(ntdll.dll, kernel32.dll, msvcrt.dll, …)
- - E_R.handler doesn't pointers stack area.
- - E_R.handler only can pointers one of registered exception handler addresses or unloaded module address range.
- SEHOP protection:
- - SEH chain(E_R.Next -> E_R.Next -> …) must be never corrupted.
- - E_R struct address are must 4byte aligned.
- - new final handler after default handler(kernel32!_except_handler*) added on SEHOP applied platforms also must be never currupted.
- - win server 2008, win server 2008 R2 (default enabled)
- - win vista sp1 also supported but disabled by default.

6. History of SEH overflow

- SafeSEH bypass method:
  - ‘Defeating the Stack Based Buffer Overflow Prevention Mechanism of
  - Microsoft Windows 2003 Server.’
  - (http://www.ngssoftware.com/papers/defeating-w2k3-stack-protection.pdf)
  - David Litchfield, 8 September 2003.
  - - kind of approaches of SEH overflows

- best SEH exploit writing tutorials:
  - Corelan Team (http://www.corelan.be:8800/index.php/articles/)

- SEHOP bypass method:
  - ‘Bypassing SEHOP’ (http://www.sysdream.com/articles/sehop_en.pdf)
  - - SYSDREAM Cooperation, lastly (2010 ?!)

7. SEH All-at-Once attack method

- new attack method to bypass SafeSEH + SEHOP protections all at once.

- SafeSEH bypass:
- Registered Exception handler(DLL or Images compiler generated _except_handler3 or same code of _except_handler3) allowed to execute! Then can attack the handler!
- SEHOP bypass:
- E_R.Next chains must pointers next valid E_R struct. (to final handler's E_R struct) Then If we can change the E_R.Next on overflowed stack frame. And saves shellcode onto vulnerable buffer than after '|RET|'.

Cont.

```
| vuln buffer                          |  E_R struct |      SFP      |    RET    |
| Shellcode     [&shellcode]           | (1)         | Crahser bytes.
      ^                       |     ^           |
      |                       |     |           +--------------------------+
      | (3)                   |     | (2)                                  |
      +------------+       +------+ ( _except_handler3 routine ) <--+
```

- (1) E_R.Next = Original value. (don’t change!)
- E_R.Handler = *allowed module!_except_handler3 address.
- E_R+8   [ebp-8] = an address of Image area.
- E_R+Ch [ebp-4] = index to user-defined exception handler for each ‘__try{‘.

Cont.

- If you cacluate properly the [ebp-8] and [ebp-4] together to pointer an address of shellcode. Then After called _except_handler3 handler, firstly calculated the user-defined exception handler address by using the two values and finally makes an indirect calling using the address.
- When the calculated address(address of &shellcode) is called indirectly... then eip register pointers the shellcode!

Cont.

- This method similar with one of David Litchfield's 2003 technique. But the method is some different at the way of execute shellcode. And it was applies only for SafeSEH. SEHOP protection was doesn't exists at the time.
- And so This presentation says a new try and new method to exploit SafeSEH+SEHOP.
- (about this, you can reference Case by case exploit's comment)

8. Case by case exploit

- AudioTran stackoverflow case by case exploit:
- http://www.x90c.org/All_at_Once_SEH_attack/audiotran_safeseh_sehop_exploit(SEH_all-at-once_attack).c.txt

9. End

- Thank you.

- x90c (Kyong Joo, Jung)
- Personal Homepage: http://www.x90c.org
- E-mail: geinblues@gmail.com