SEH overwrite and its exploitability

Fourteenforty Research Institute Inc.

Agenda

- Theme and Goal
- Review of SEH overwrites
- Protection mechanisms for SEH overwrites
- Bypassing protection mechanisms.
- Demonstration
- Conclusion

Theme and goal

Theme

- SEH overwriting is one of the major methods for exploiting Windows software.
- We already have several protection mechanisms for SEH overwrites.
- Is the protection provided by DEP and SEHOP enough?
- How about the protection from SafeSEH and SEHOP?

Goal

- To reveal which combinations of the known protection mechanisms for SEH overwrites are really effective.
- On the way to the goal, I will show you that we can bypass SafeSEH and Software DEP and Hardware DEP and SEHOP, all at the same time, under certain conditions.

Target environment

- Windows XP SP3
- Windows Vista SP1
- Windows 7

- 32bit processes on x86
- 32bit processes on x64 (WOW64)

- Visual Studio 2008 was used to compile all the programs in this presentation.

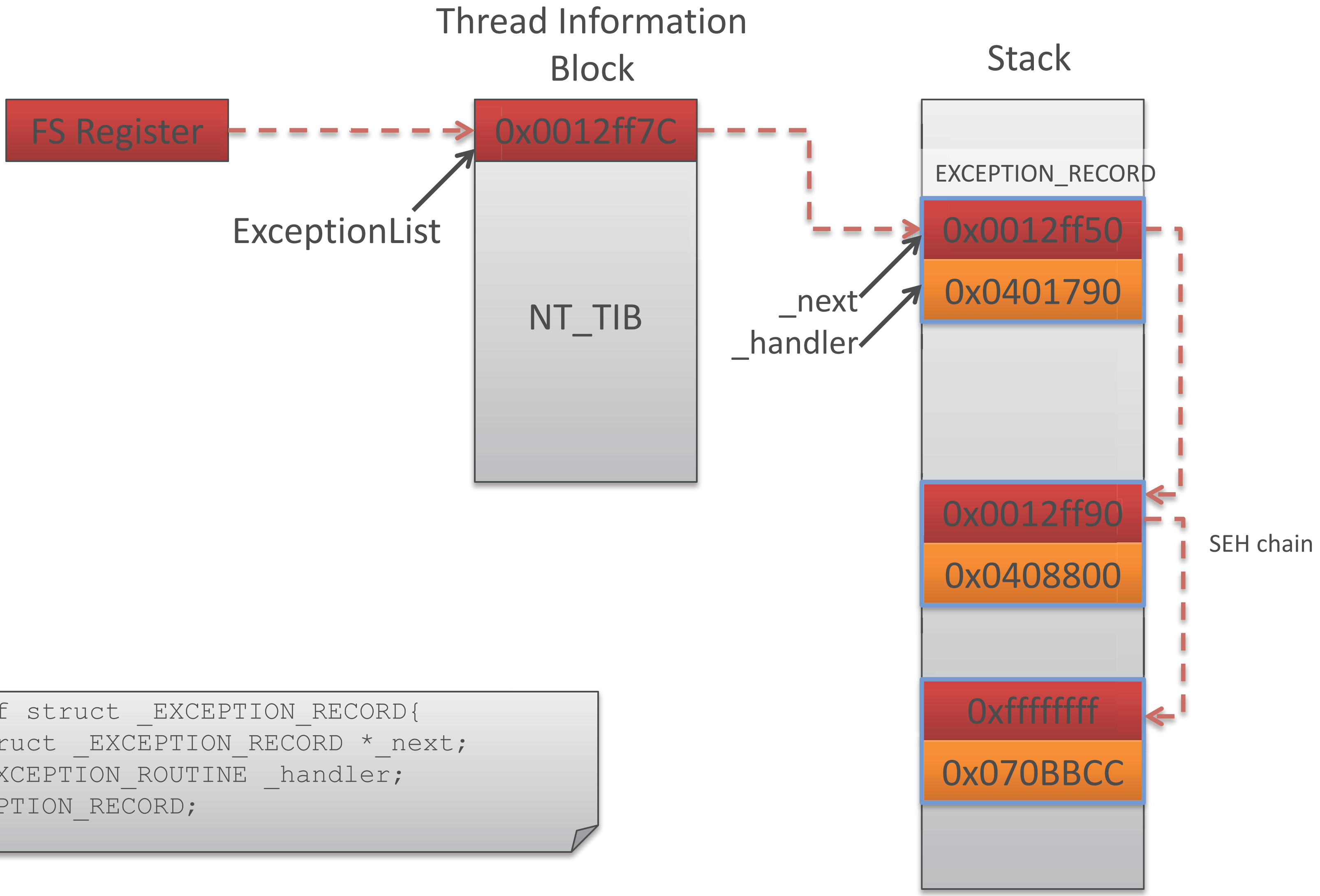
Review of SEH Overwrite

About SEH

- SEH is “Structured Exception Handling”
- Exception handling system provided by Windows

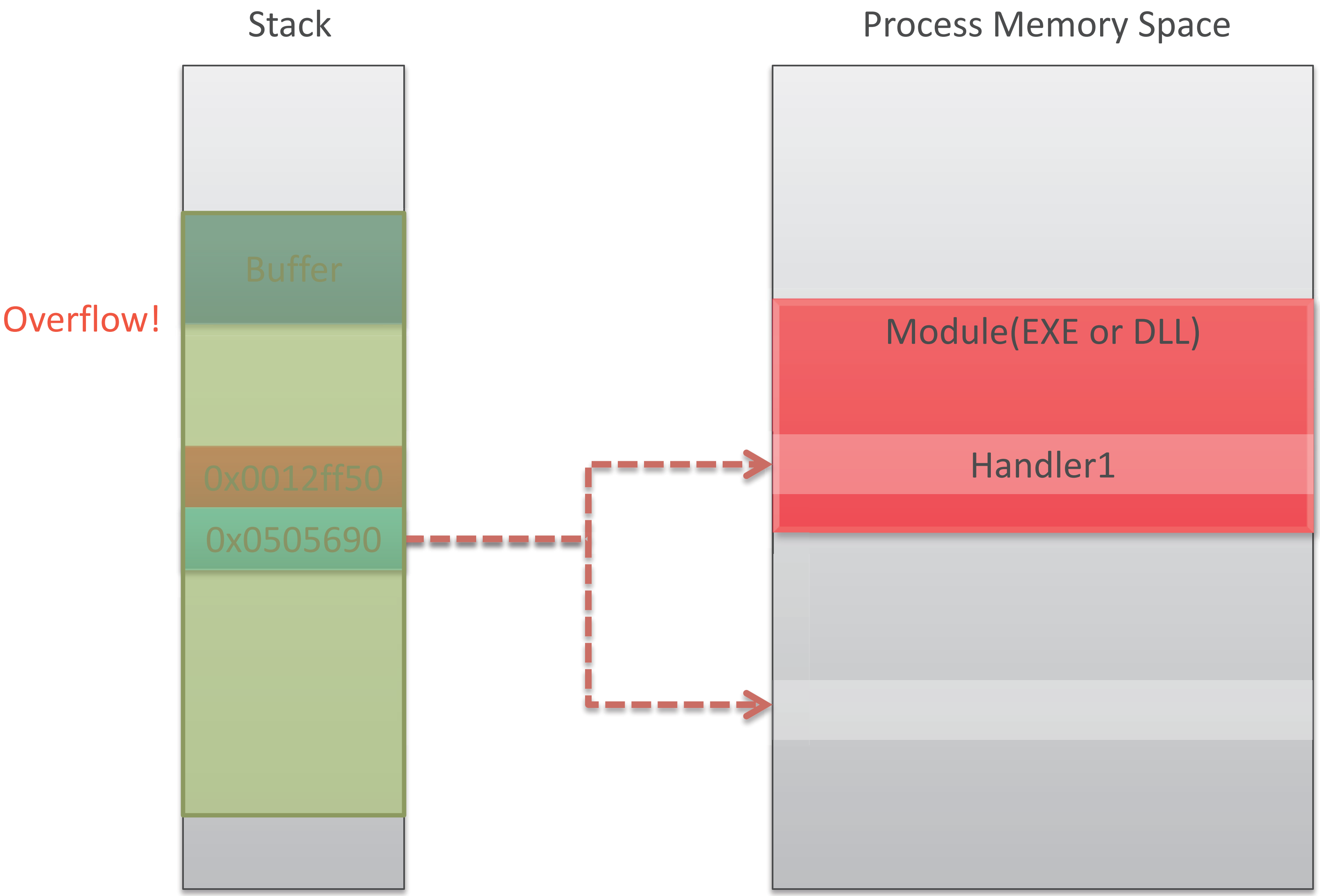
```
int test(void){
__try{
   // Exception may occur here
}
__except( EXCEPTION_EXECUTE_HANDLER ){
   // This handles the exception
}

return 0;
}
```

Inside Windows Exception Dispatching

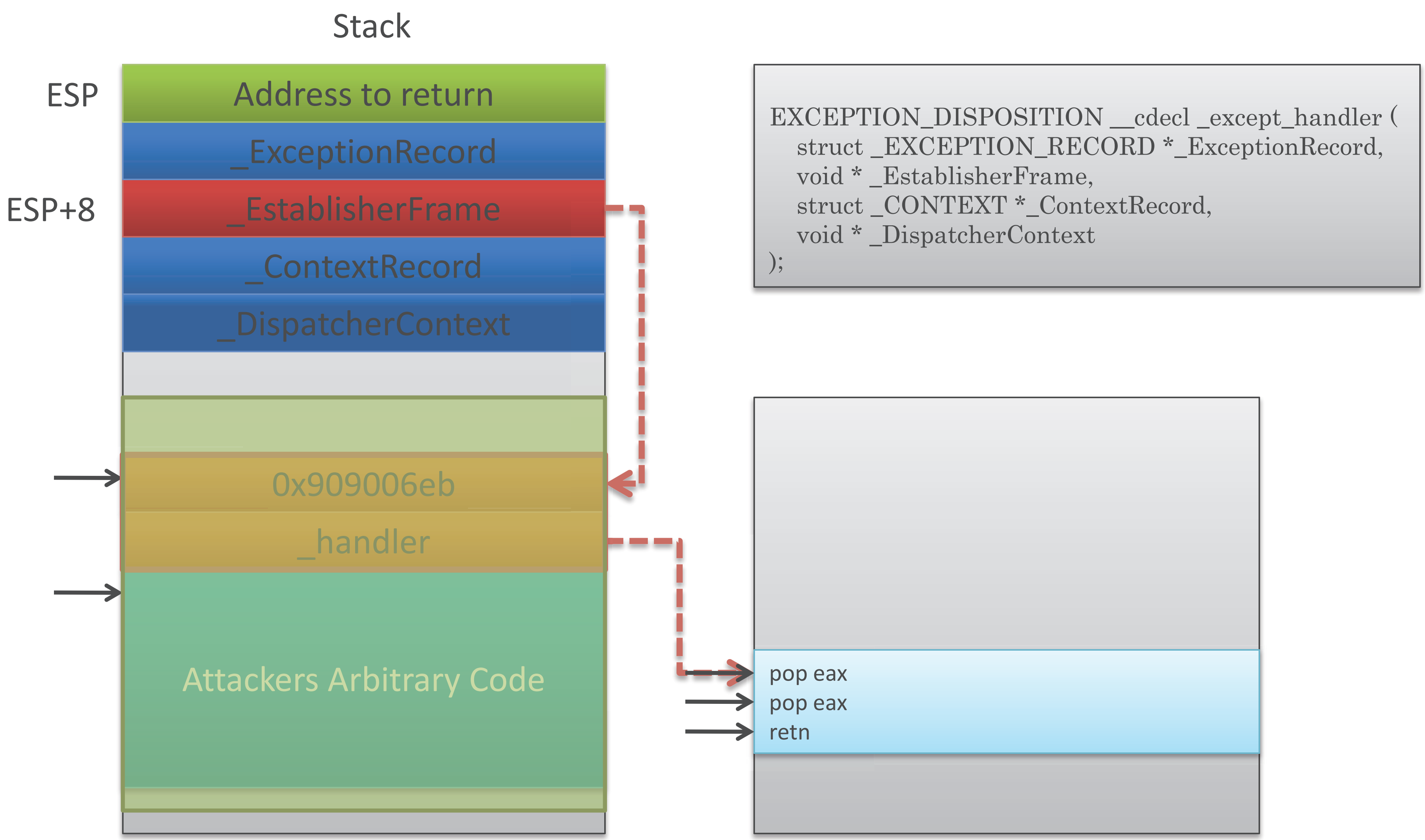


```
typedef struct _EXCEPTION_RECORD{
    struct _EXCEPTION_RECORD *_next;
    PEXCEPTION_ROUTINE _handler;
} EXCEPTION_RECORD;
```

SEH Overwrite Attack



Stack after an exception handler is called



```
EXCEPTION_DISPOSITION __cdecl _except_handler (
    struct _EXCEPTION_RECORD *_ExceptionRecord,
    void * _EstablisherFrame,
    struct _CONTEXT *_ContextRecord,
    void * _DispatcherContext
);
```

```
pop eax
pop eax
retn
```

Protection mechanisms

Protection mechanisms

- /GS
- SafeSEH
- SoftwareDEP
- SEHOP
- Hardware DEP
- ASLR

/GS

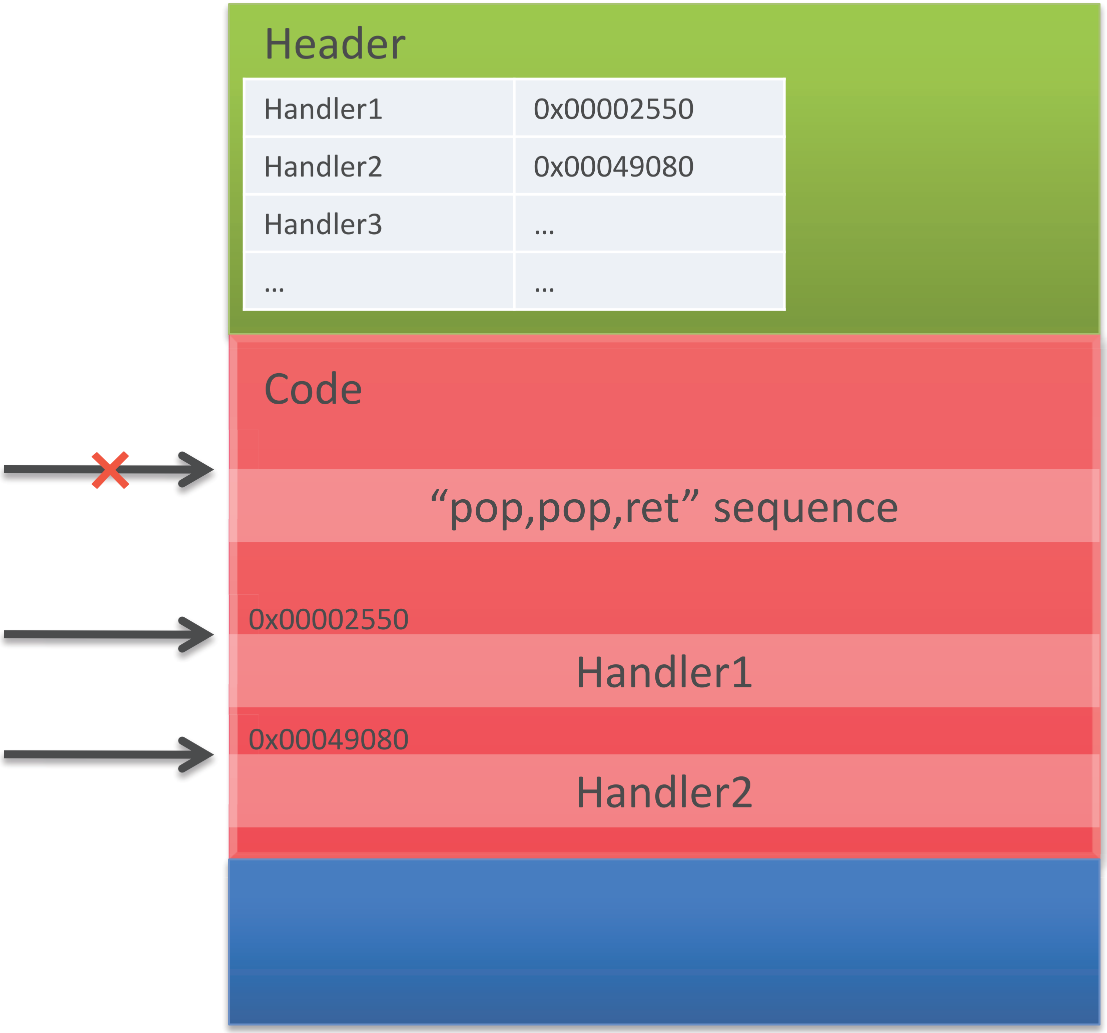
- Stack Guard by Visual Studio compiler
- This checks if a buffer overflow occurs before returning from a function.
- This is irrelevant to SEH overwrites because an exception can be generated before the check.

Protection mechanisms

SEH overwrites specific

SafeSEH

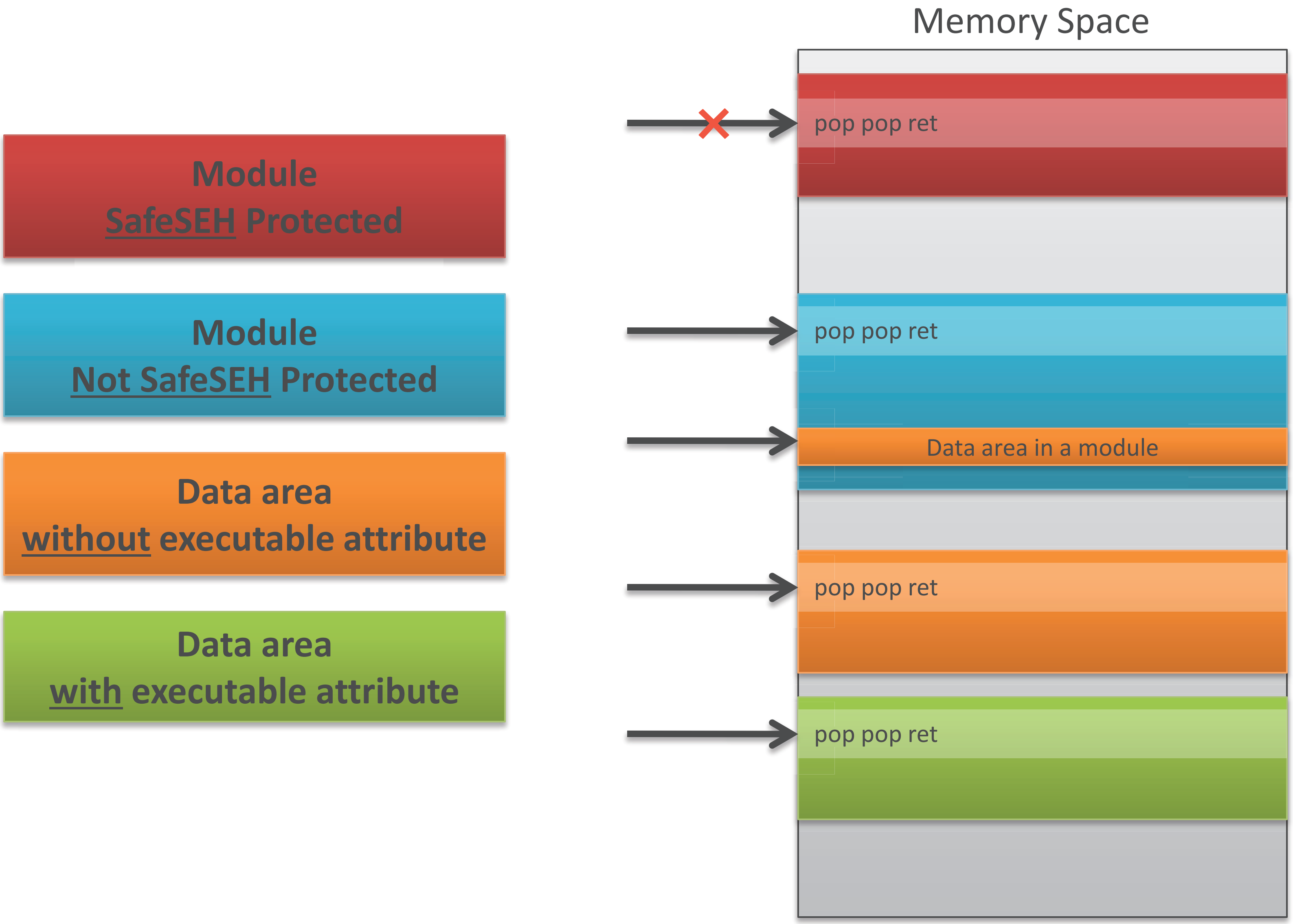
A module protected by SafeSEH



Header

Handler1	0x00002550
Handler2	0x00049080
Handler3	…
…	…

Weakness of SafeSEH



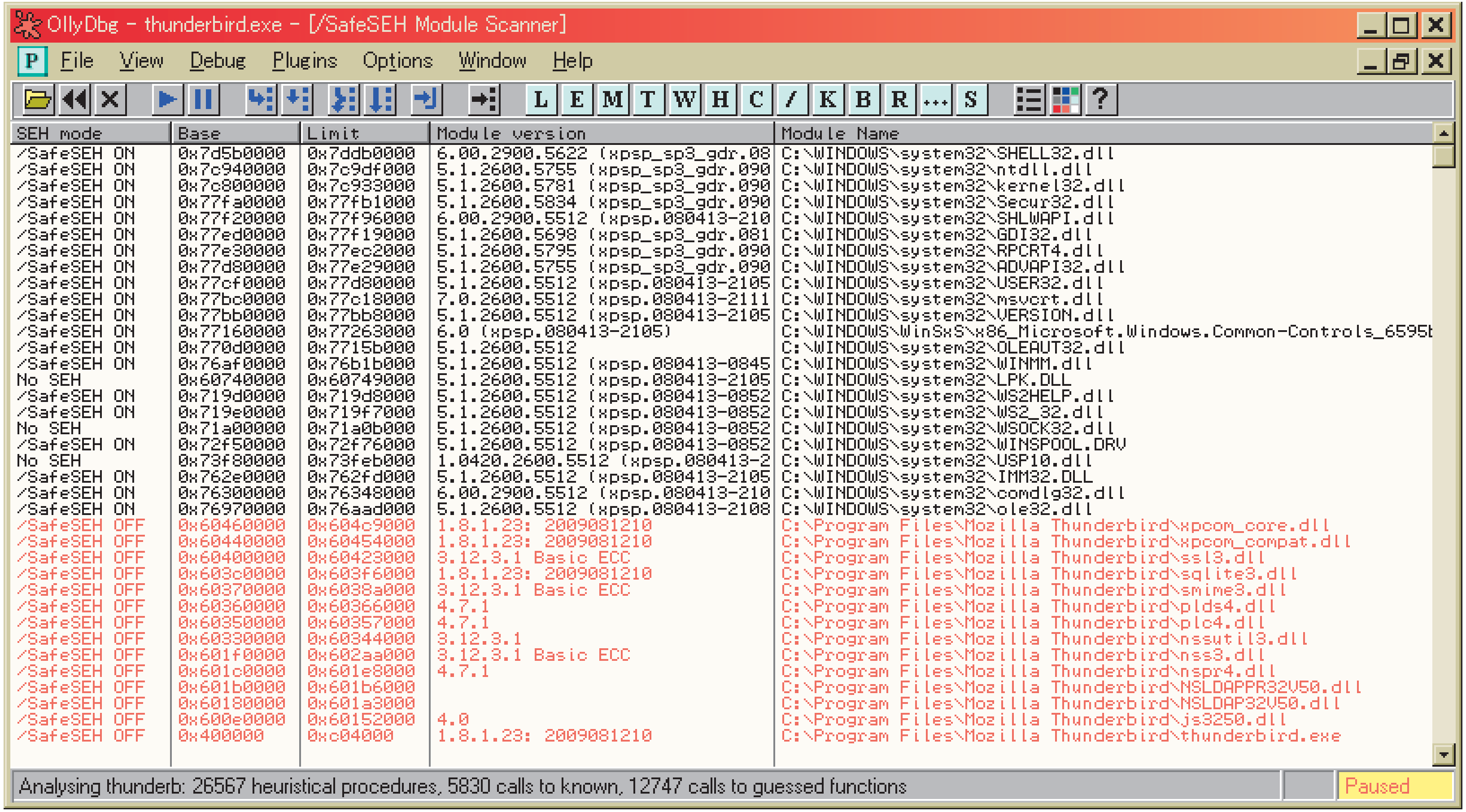
Module
SafeSEH Protected

Module
Not SafeSEH Protected

Data area
without executable attribute

Data area
with executable attribute

Modules not protected by SafeSEH


OllyDbg - thunderbird.exe - [/SafeSEH Module Scanner]
P File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?
Paused

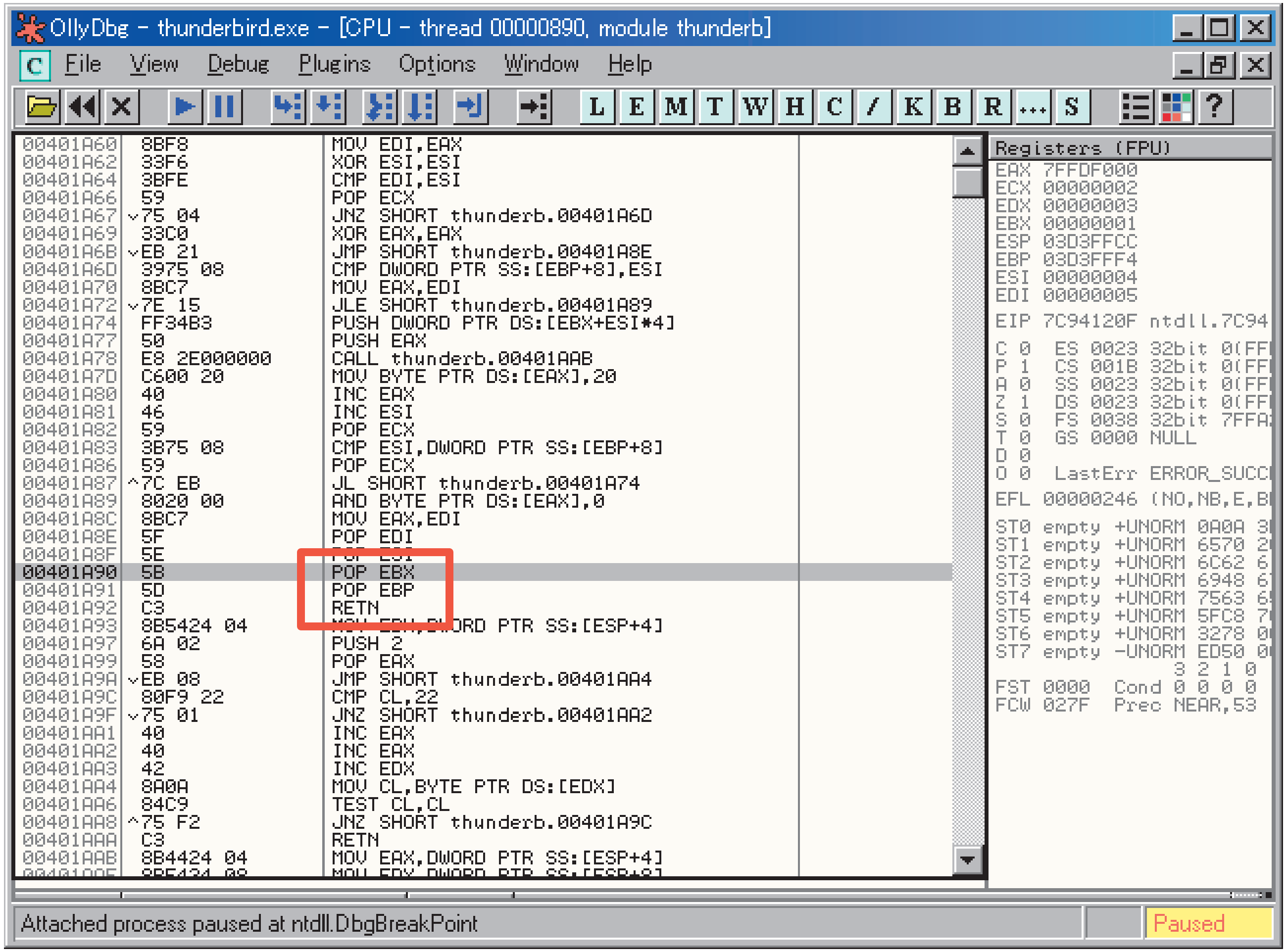
```
SEH mode        Base        Limit       Module version                Module Name
/SafeSEH ON     0x7d5b0000  0x7ddb0000  6.00.2900.5622 (xpsp_sp3_gdr.08  C:\WINDOWS\system32\SHELL32.dll
/SafeSEH ON     0x7c940000  0x7c9df000  5.1.2600.5755 (xpsp_sp3_gdr.090  C:\WINDOWS\system32\ntdll.dll
/SafeSEH ON     0x7c800000  0x7c933000  5.1.2600.5781 (xpsp_sp3_gdr.090  C:\WINDOWS\system32\kernel32.dll
/SafeSEH ON     0x77fa0000  0x77fb1000  5.1.2600.5834 (xpsp_sp3_gdr.090  C:\WINDOWS\system32\Secur32.dll
/SafeSEH ON     0x77f20000  0x77f96000  6.00.2900.5512 (xpsp.080413-210  C:\WINDOWS\system32\SHLWAPI.dll
/SafeSEH ON     0x77ed0000  0x77f19000  5.1.2600.5698 (xpsp_sp3_gdr.081  C:\WINDOWS\system32\GDI32.dll
/SafeSEH ON     0x77e30000  0x77ec2000  5.1.2600.5795 (xpsp_sp3_gdr.090  C:\WINDOWS\system32\RPCRT4.dll
/SafeSEH ON     0x77d80000  0x77e29000  5.1.2600.5755 (xpsp_sp3_gdr.090  C:\WINDOWS\system32\ADVAPI32.dll
/SafeSEH ON     0x77cf0000  0x77d80000  5.1.2600.5512 (xpsp.080413-2105  C:\WINDOWS\system32\USER32.dll
/SafeSEH ON     0x77bc0000  0x77c18000  7.0.2600.5512 (xpsp.080413-2111  C:\WINDOWS\system32\msvcrt.dll
/SafeSEH ON     0x77bb0000  0x77bb8000  5.1.2600.5512 (xpsp.080413-2105  C:\WINDOWS\system32\VERSION.dll
/SafeSEH ON     0x77160000  0x77263000  6.0 (xpsp.080413-2105)           C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b
/SafeSEH ON     0x770d0000  0x7715b000  5.1.2600.5512                    C:\WINDOWS\system32\OLEAUT32.dll
/SafeSEH ON     0x76af0000  0x76b1b000  5.1.2600.5512 (xpsp.080413-0845  C:\WINDOWS\system32\WINMM.dll
No SEH          0x60740000  0x60749000  5.1.2600.5512 (xpsp.080413-2105  C:\WINDOWS\system32\LPK.DLL
/SafeSEH ON     0x719d0000  0x719d8000  5.1.2600.5512 (xpsp.080413-0852  C:\WINDOWS\system32\WS2HELP.dll
/SafeSEH ON     0x719e0000  0x719f7000  5.1.2600.5512 (xpsp.080413-0852  C:\WINDOWS\system32\WS2_32.dll
No SEH          0x71a00000  0x71a0b000  5.1.2600.5512 (xpsp.080413-0852  C:\WINDOWS\system32\WSOCK32.dll
/SafeSEH ON     0x72f50000  0x72f76000  5.1.2600.5512 (xpsp.080413-0852  C:\WINDOWS\system32\WINSPOOL.DRV
No SEH          0x73f80000  0x73feb000  1.0420.2600.5512 (xpsp.080413-2  C:\WINDOWS\system32\USP10.dll
/SafeSEH ON     0x762e0000  0x762fd000  5.1.2600.5512 (xpsp.080413-2105  C:\WINDOWS\system32\IMM32.DLL
/SafeSEH ON     0x76300000  0x76348000  6.00.2900.5512 (xpsp.080413-210  C:\WINDOWS\system32\comdlg32.dll
/SafeSEH ON     0x76970000  0x76aad000  5.1.2600.5512 (xpsp.080413-2108  C:\WINDOWS\system32\ole32.dll
/SafeSEH OFF    0x60460000  0x604c9000  1.8.1.23: 2009081210             C:\Program Files\Mozilla Thunderbird\xpcom_core.dll
/SafeSEH OFF    0x60440000  0x60454000  1.8.1.23: 2009081210             C:\Program Files\Mozilla Thunderbird\xpcom_compat.dll
/SafeSEH OFF    0x60400000  0x60423000  3.12.3.1 Basic ECC               C:\Program Files\Mozilla Thunderbird\ssl3.dll
/SafeSEH OFF    0x603c0000  0x603f6000  1.8.1.23: 2009081210             C:\Program Files\Mozilla Thunderbird\sqlite3.dll
/SafeSEH OFF    0x60370000  0x6038a000  3.12.3.1 Basic ECC               C:\Program Files\Mozilla Thunderbird\smime3.dll
/SafeSEH OFF    0x60360000  0x60366000  4.7.1                            C:\Program Files\Mozilla Thunderbird\plds4.dll
/SafeSEH OFF    0x60350000  0x60357000  4.7.1                            C:\Program Files\Mozilla Thunderbird\plc4.dll
/SafeSEH OFF    0x60330000  0x60344000  3.12.3.1                         C:\Program Files\Mozilla Thunderbird\nssutil3.dll
/SafeSEH OFF    0x601f0000  0x602aa000  3.12.3.1 Basic ECC               C:\Program Files\Mozilla Thunderbird\nss3.dll
/SafeSEH OFF    0x601c0000  0x601e8000  4.7.1                            C:\Program Files\Mozilla Thunderbird\nspr4.dll
/SafeSEH OFF    0x601b0000  0x601b6000                                   C:\Program Files\Mozilla Thunderbird\NSLDAPPR32V50.dll
/SafeSEH OFF    0x60180000  0x601a3000                                   C:\Program Files\Mozilla Thunderbird\NSLDAP32V50.dll
/SafeSEH OFF    0x600e0000  0x60152000  4.0                              C:\Program Files\Mozilla Thunderbird\js3250.dll
/SafeSEH OFF    0x400000    0xc04000    1.8.1.23: 2009081210             C:\Program Files\Mozilla Thunderbird\thunderbird.exe
```

Analysing thunderb: 26567 heuristical procedures, 5830 calls to known, 12747 calls to guessed functions

From Thunderbird 2.0.0.23

* All modules in Thunderbird 3.0.3 are compiled with /SafeSEH

pop pop ret in the wild


OllyDbg - thunderbird.exe - [CPU - thread 00000890, module thunderb]
C
File
View
Debug
Plugins
Options
Window
Help
L
E
M
T
W
H
C
/
K
B
R
...
S
?

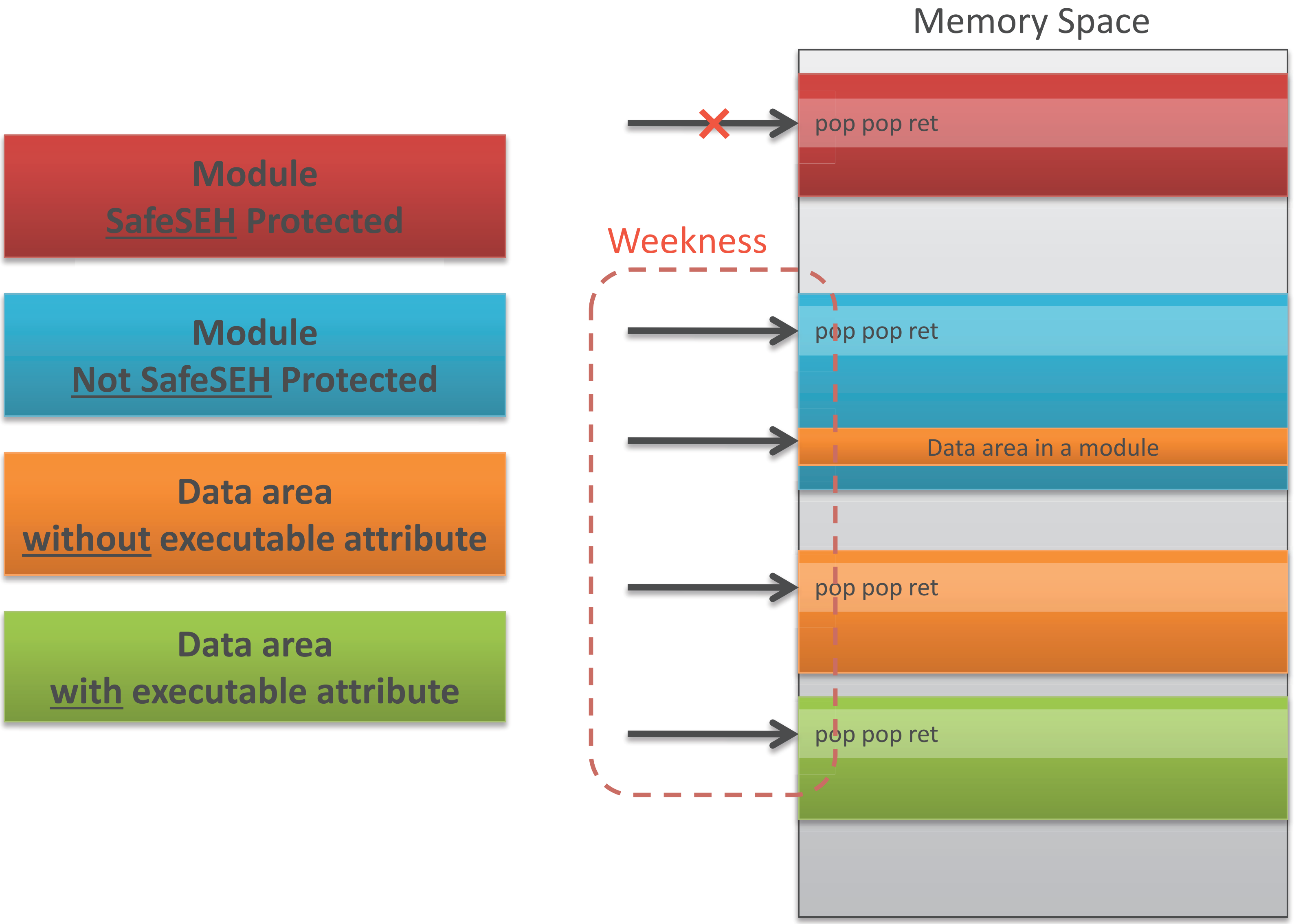
```
00401A60   8BF8         MOV EDI,EAX
00401A62   33F6         XOR ESI,ESI
00401A64   3BFE         CMP EDI,ESI
00401A66   59           POP ECX
00401A67  v75 04        JNZ SHORT thunderb.00401A6D
00401A69   33C0         XOR EAX,EAX
00401A6B  vEB 21        JMP SHORT thunderb.00401A8E
00401A6D   3975 08      CMP DWORD PTR SS:[EBP+8],ESI
00401A70   8BC7         MOV EAX,EDI
00401A72  v7E 15        JLE SHORT thunderb.00401A89
00401A74   FF34B3       PUSH DWORD PTR DS:[EBX+ESI*4]
00401A77   50           PUSH EAX
00401A78   E8 2E000000  CALL thunderb.00401AAB
00401A7D   C600 20      MOV BYTE PTR DS:[EAX],20
00401A80   40           INC EAX
00401A81   46           INC ESI
00401A82   59           POP ECX
00401A83   3B75 08      CMP ESI,DWORD PTR SS:[EBP+8]
00401A86   59           POP ECX
00401A87  ^7C EB        JL SHORT thunderb.00401A74
00401A89   8020 00      AND BYTE PTR DS:[EAX],0
00401A8C   8BC7         MOV EAX,EDI
00401A8E   5F           POP EDI
00401A8F   5E           POP ESI
00401A90   5B           POP EBX
00401A91   5D           POP EBP
00401A92   C3           RETN
00401A93   8B5424 04    MOV EDX,DWORD PTR SS:[ESP+4]
00401A97   6A 02        PUSH 2
00401A99   58           POP EAX
00401A9A  vEB 08        JMP SHORT thunderb.00401AA4
00401A9C   80F9 22      CMP CL,22
00401A9F  v75 01        JNZ SHORT thunderb.00401AA2
00401AA1   40           INC EAX
00401AA2   40           INC EAX
00401AA3   42           INC EDX
00401AA4   8A0A         MOV CL,BYTE PTR DS:[EDX]
00401AA6   84C9         TEST CL,CL
00401AA8  ^75 F2        JNZ SHORT thunderb.00401A9C
00401AAA   C3           RETN
00401AAB   8B4424 04    MOV EAX,DWORD PTR SS:[ESP+4]
00401AAF   8B5424 08    MOV EDX,DWORD PTR SS:[ESP+8]
```

```
Registers (FPU)
EAX 7FFDF000
ECX 00000002
EDX 00000003
EBX 00000001
ESP 03D3FFCC
EBP 03D3FFF4
ESI 00000004
EDI 00000005

EIP 7C94120F ntdll.7C94

C 0  ES 0023 32bit 0(FF
P 1  CS 001B 32bit 0(FF
A 0  SS 0023 32bit 0(FF
Z 1  DS 0023 32bit 0(FF
S 0  FS 0038 32bit 7FFA
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCC

EFL 00000246 (NO,NB,E,B

ST0 empty +UNORM 0A0A 3
ST1 empty +UNORM 6570 2
ST2 empty +UNORM 6C62 6
ST3 empty +UNORM 6948 6
ST4 empty +UNORM 7563 6
ST5 empty +UNORM 5FC8 7
ST6 empty +UNORM 3278 0
ST7 empty -UNORM ED50 0
                3 2 1 0
FST 0000  Cond 0 0 0 0
FCW 027F  Prec NEAR,53
```

```
Attached process paused at ntdll.DbgBreakPoint                Paused
```

From thunderbird.exe

Additional check by Software DEP



Module
SafeSEH Protected

Module
Not SafeSEH Protected

Data area
without executable attribute

Data area
with executable attribute

Memory Space

pop pop ret

Weekness

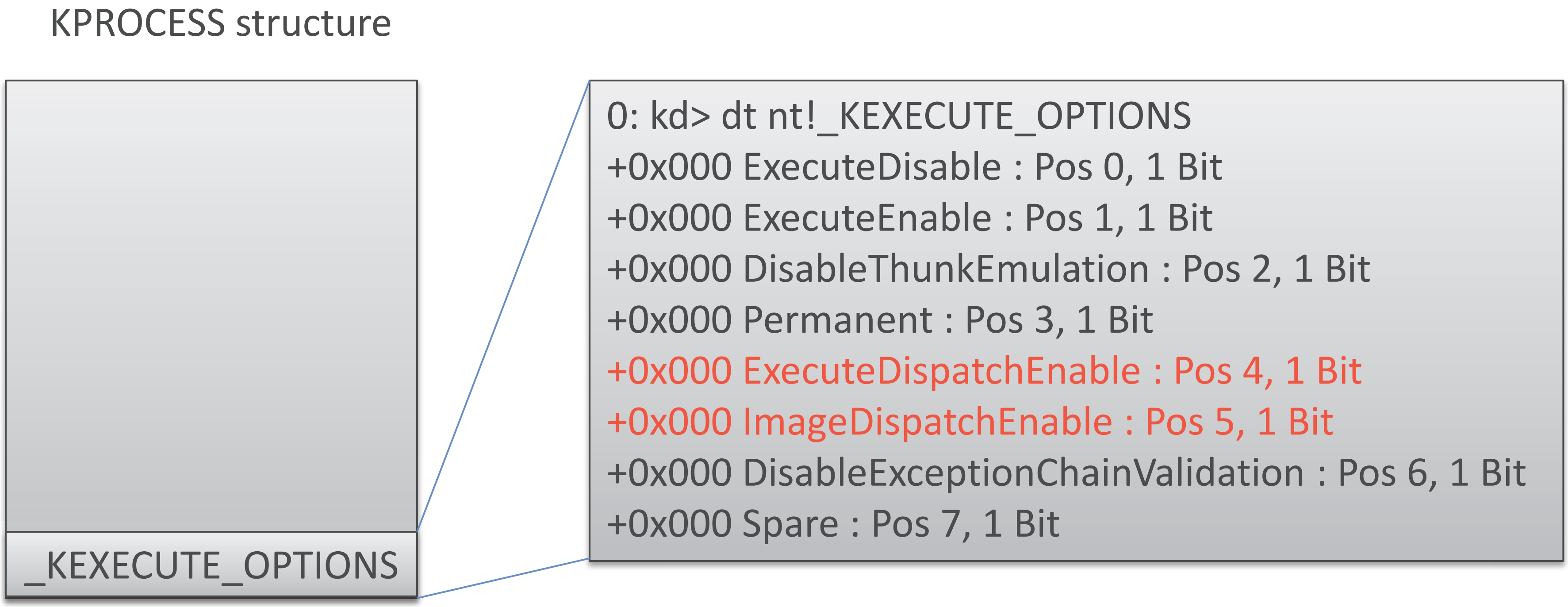
pop pop ret

Data area in a module

pop pop ret

pop pop ret

Software DEP



KPROCESS structure

_KEXECUTE_OPTIONS

```
0: kd> dt nt!_KEXECUTE_OPTIONS
+0x000 ExecuteDisable : Pos 0, 1 Bit
+0x000 ExecuteEnable : Pos 1, 1 Bit
+0x000 DisableThunkEmulation : Pos 2, 1 Bit
+0x000 Permanent : Pos 3, 1 Bit
+0x000 ExecuteDispatchEnable : Pos 4, 1 Bit
+0x000 ImageDispatchEnable : Pos 5, 1 Bit
+0x000 DisableExceptionChainValidation : Pos 6, 1 Bit
+0x000 Spare : Pos 7, 1 Bit
```

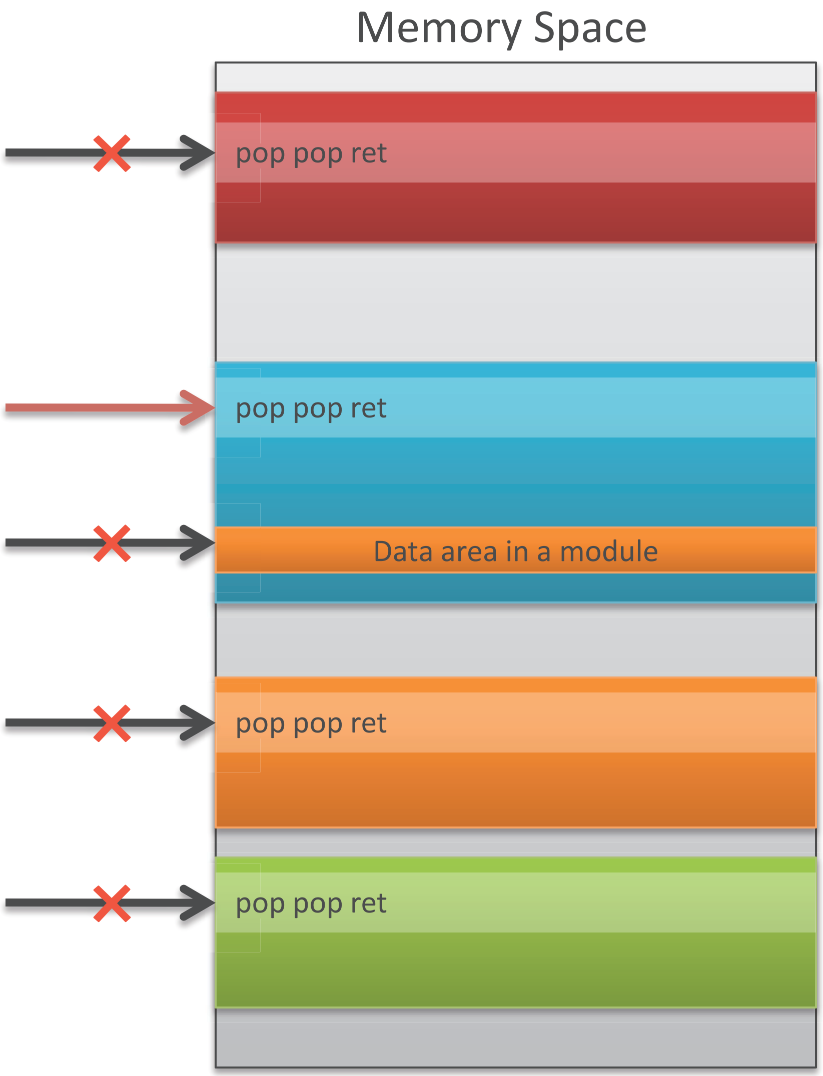
Software DEP

🟥 Module
SafeSEH Protected

🟦 Module
Not SafeSEH Protected

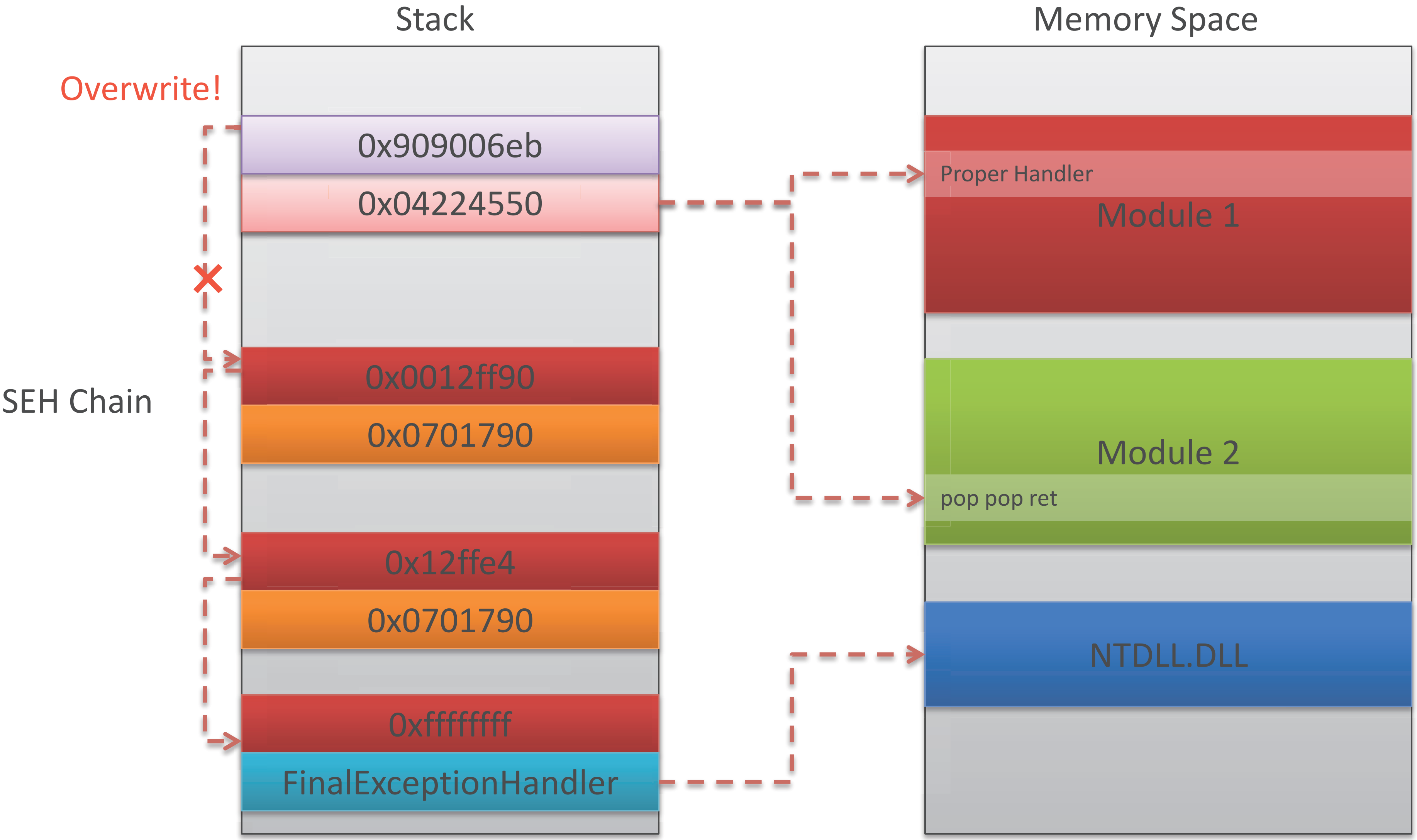
🟧 Data area
without executable attribute

🟩 Data area
with executable attribute



SEHOP

SEH overwrites protection



Protection mechanisms

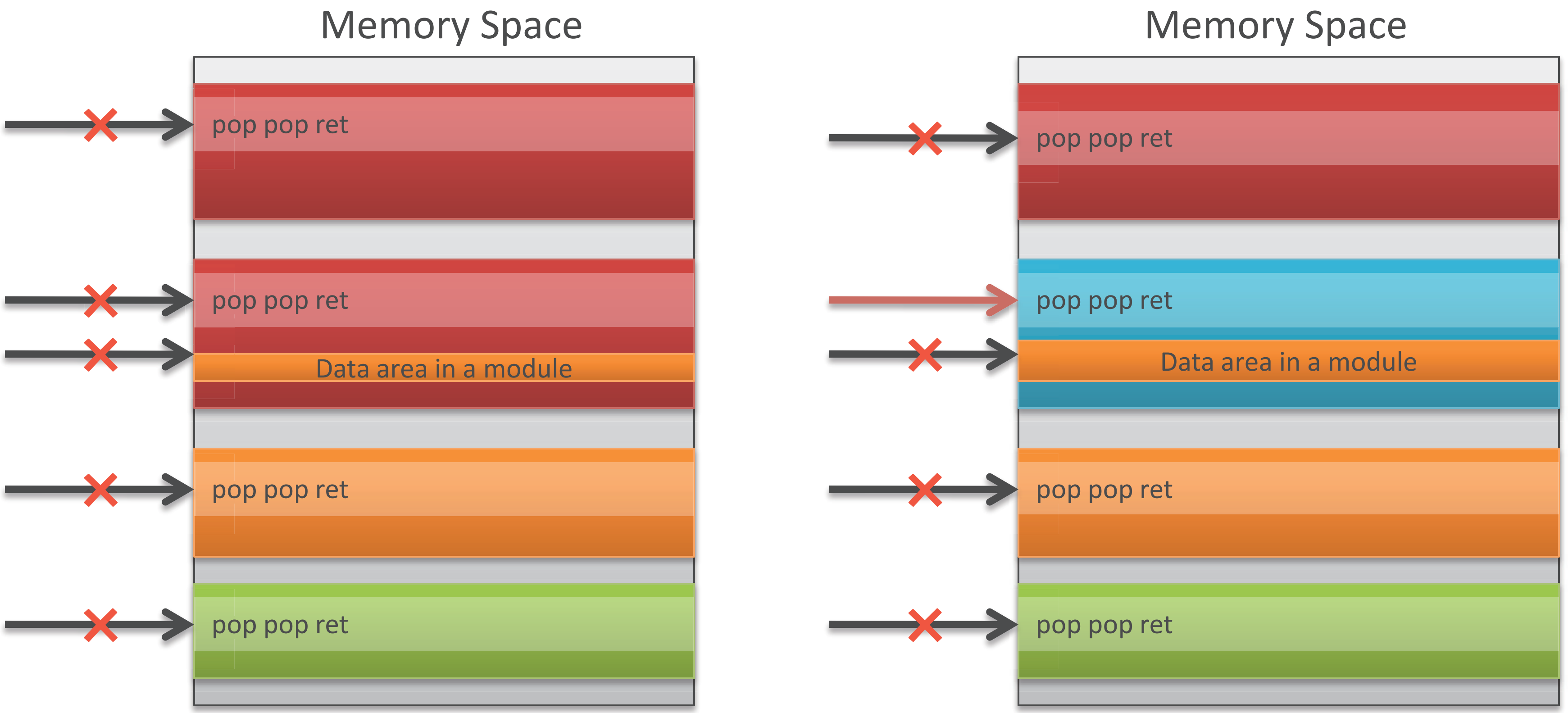
Generic protections

Hardware DEP and ASLR

- Hardware DEP prevents code without executable attribute from being executed.
- ASLR has several impacts on the SEH overwrites.(Explain later)

Bypassing protection mechanisms

SafeSEH and Software DEP



🟥 Module SafeSEH Protected

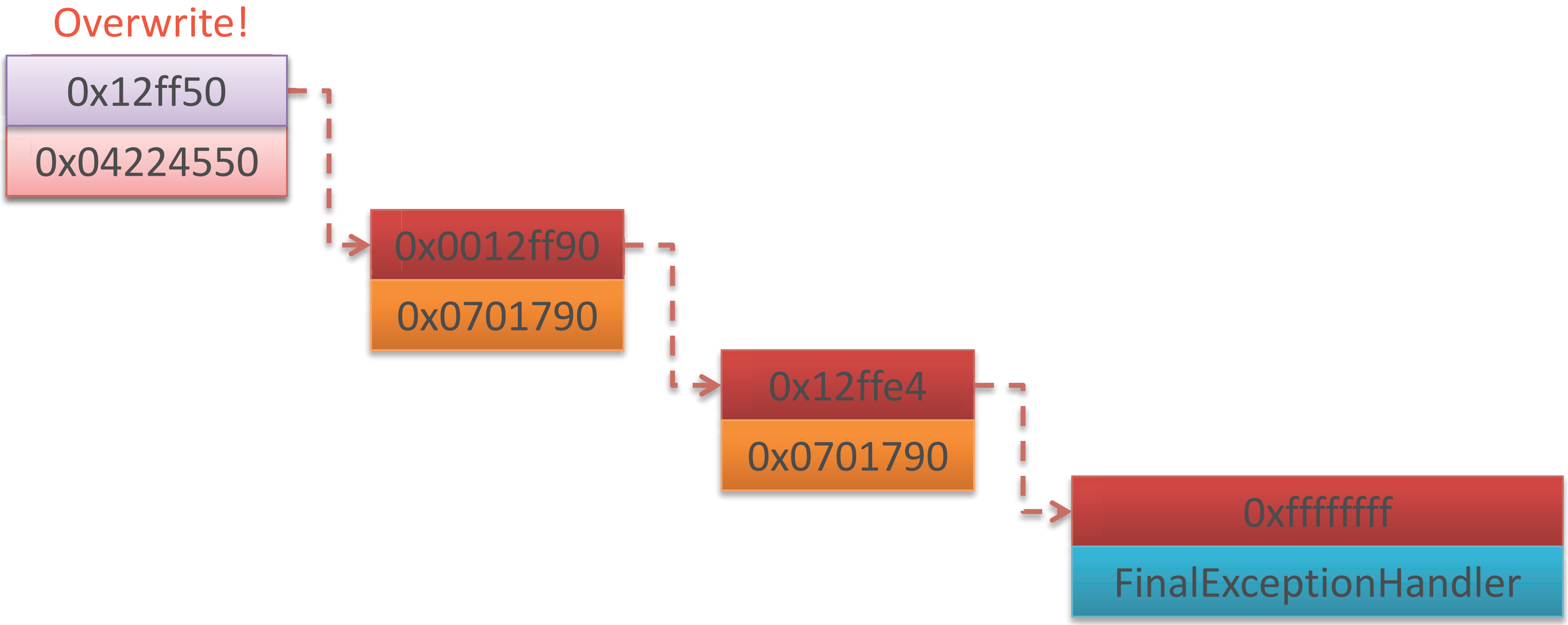
🟦 Module Not SafeSEH Protected

🟧 Non module area (Data area)

🟩 Non module area with executable attribute

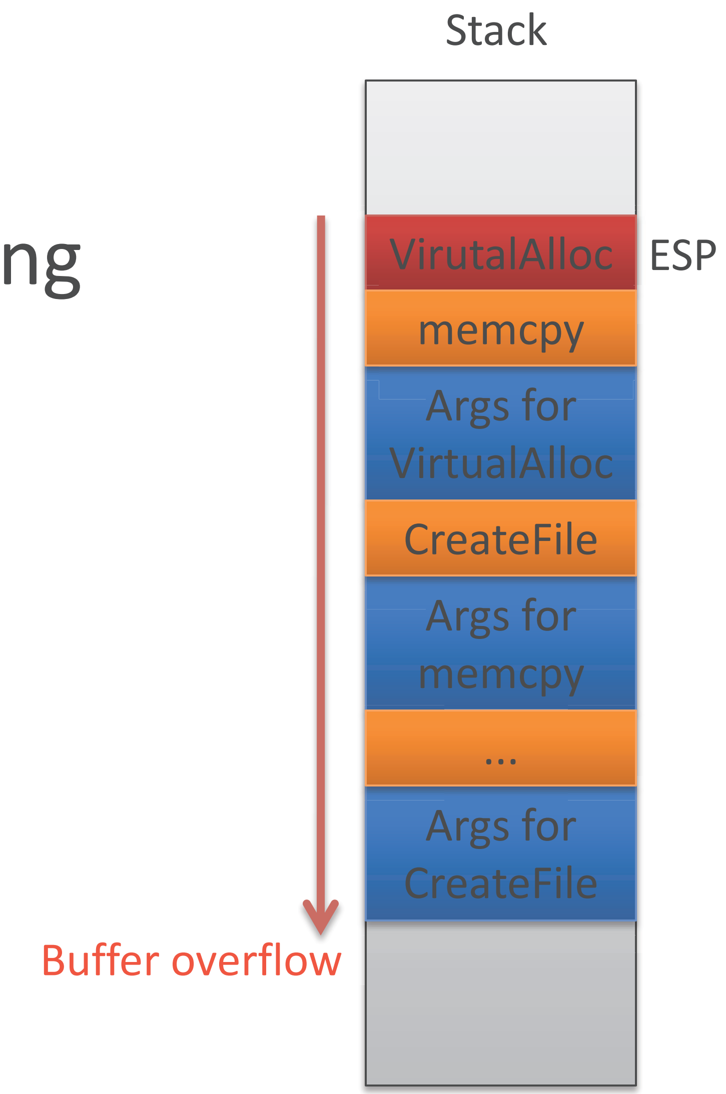
Bypassing SEHOP

- It is weak if the address of a buffer overflow and FinalExceptionHandler is known.
- Attackers can recreate a proper SEH chain.



Bypassing Hardware DEP

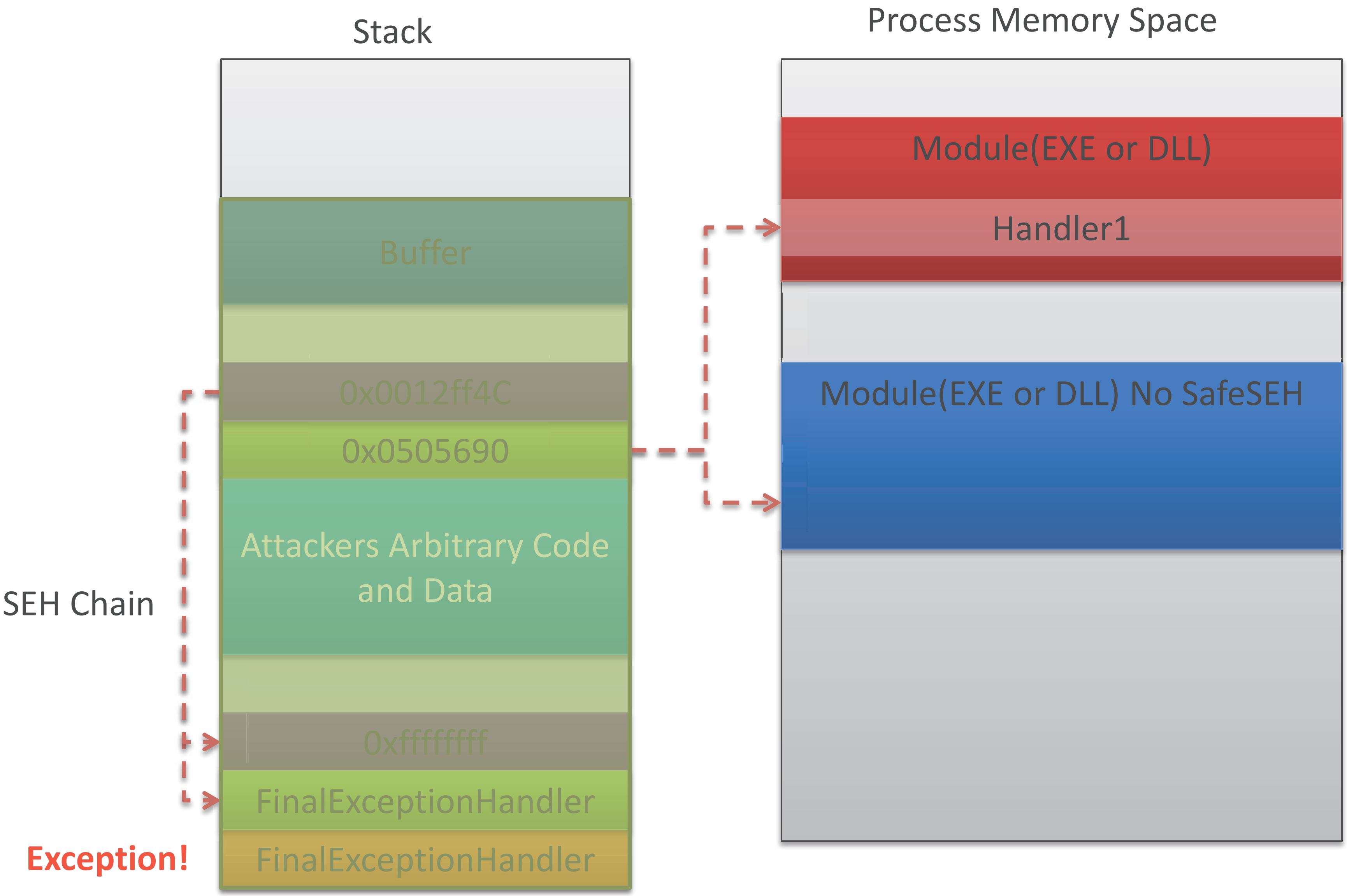
- Return-into-libc
- Return-oriented programming



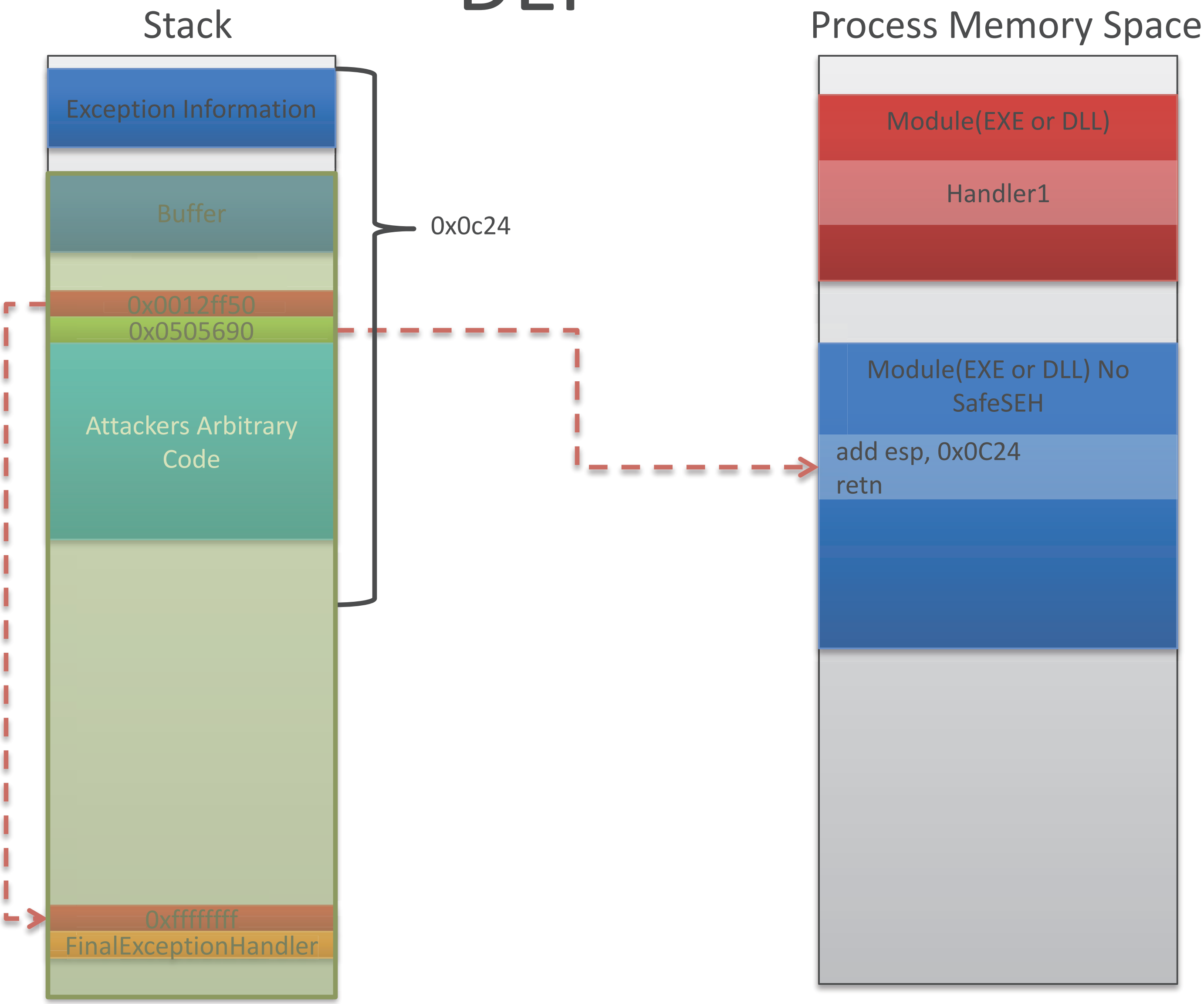
Bypass /GS, SafeSEH, Software DEP, Hardware DEP, SEHOP

- Recreate SEH Chain (bypassing SEHOP)
- Overwritten exception handler address must be in a module which is not SafeSEH enabled (bypassing SafeSEH)
- Create a stack to execute desired code (bypassing Hardware DEP)
- To execute the code using the stack above, we have to set an exception handler to some stack rewind and return instructions (bypassing Hardware DEP)
- Trigger an exception

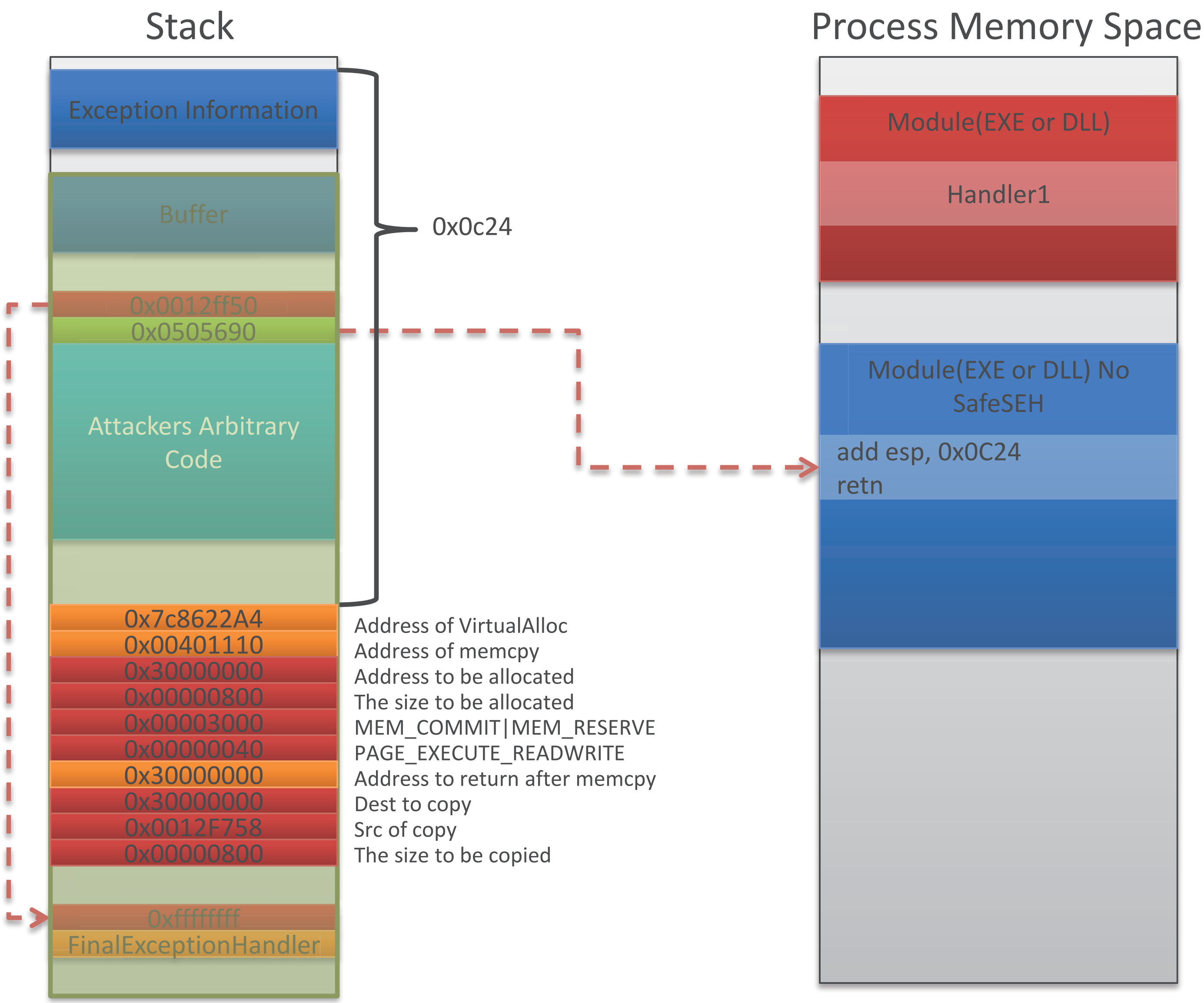
Recreating the SEH Chain and Setting the Exception Handler Address



Good instruction to bypass Hardware DEP



Create proper stack for return-into-libc



Stack

Exception Information

Buffer

0x0012ff50

0x0505690

Attackers Arbitrary Code

0x0c24

0x7c8622A4	Address of VirtualAlloc
0x00401110	Address of memcpy
0x30000000	Address to be allocated
0x00000800	The size to be allocated
0x00003000	MEM_COMMIT|MEM_RESERVE
0x00000040	PAGE_EXECUTE_READWRITE
0x30000000	Address to return after memcpy
0x30000000	Dest to copy
0x0012F758	Src of copy
0x00000800	The size to be copied

0xffffffff

FinalExceptionHandler

Process Memory Space

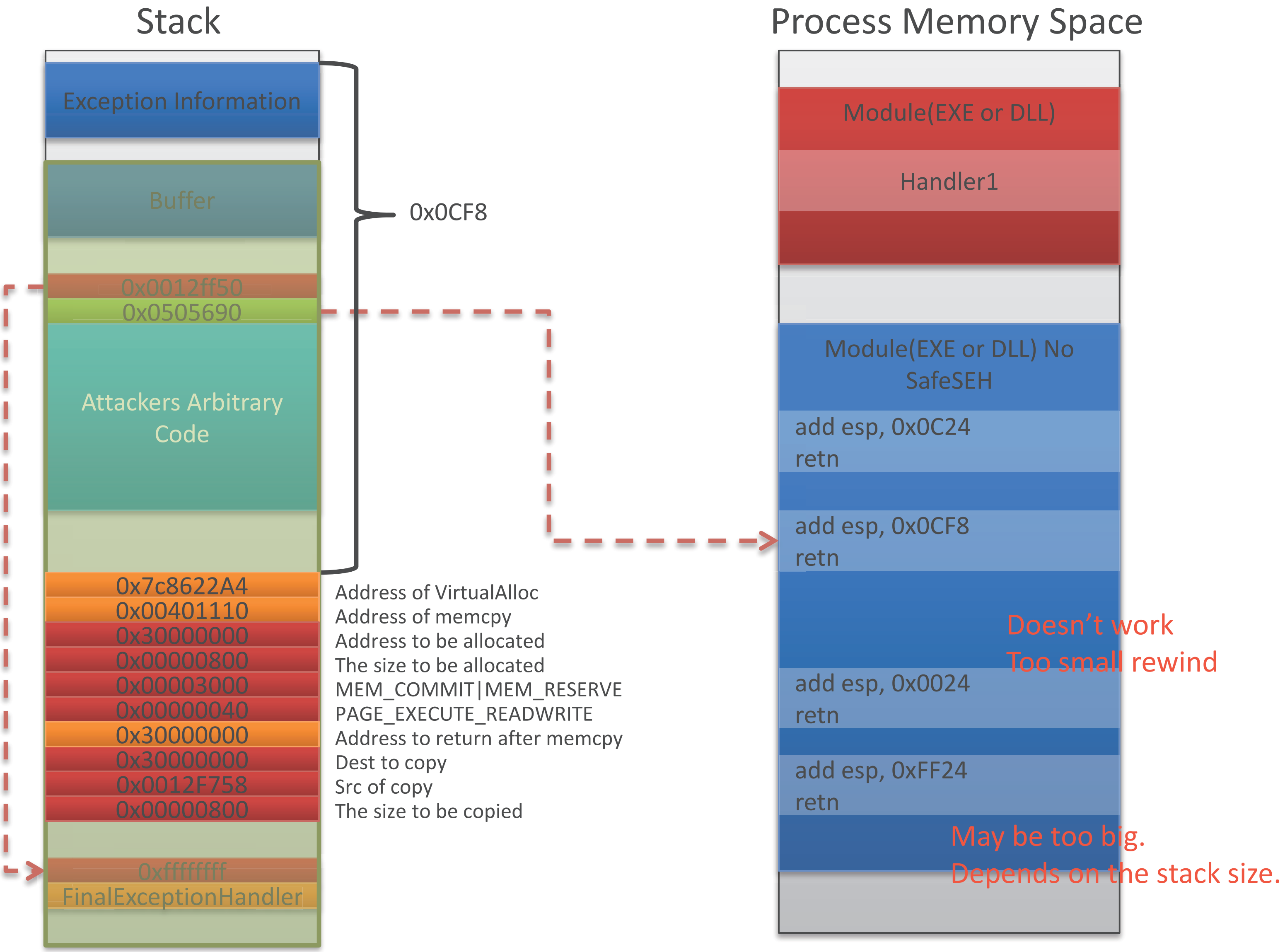
Module(EXE or DLL)

Handler1

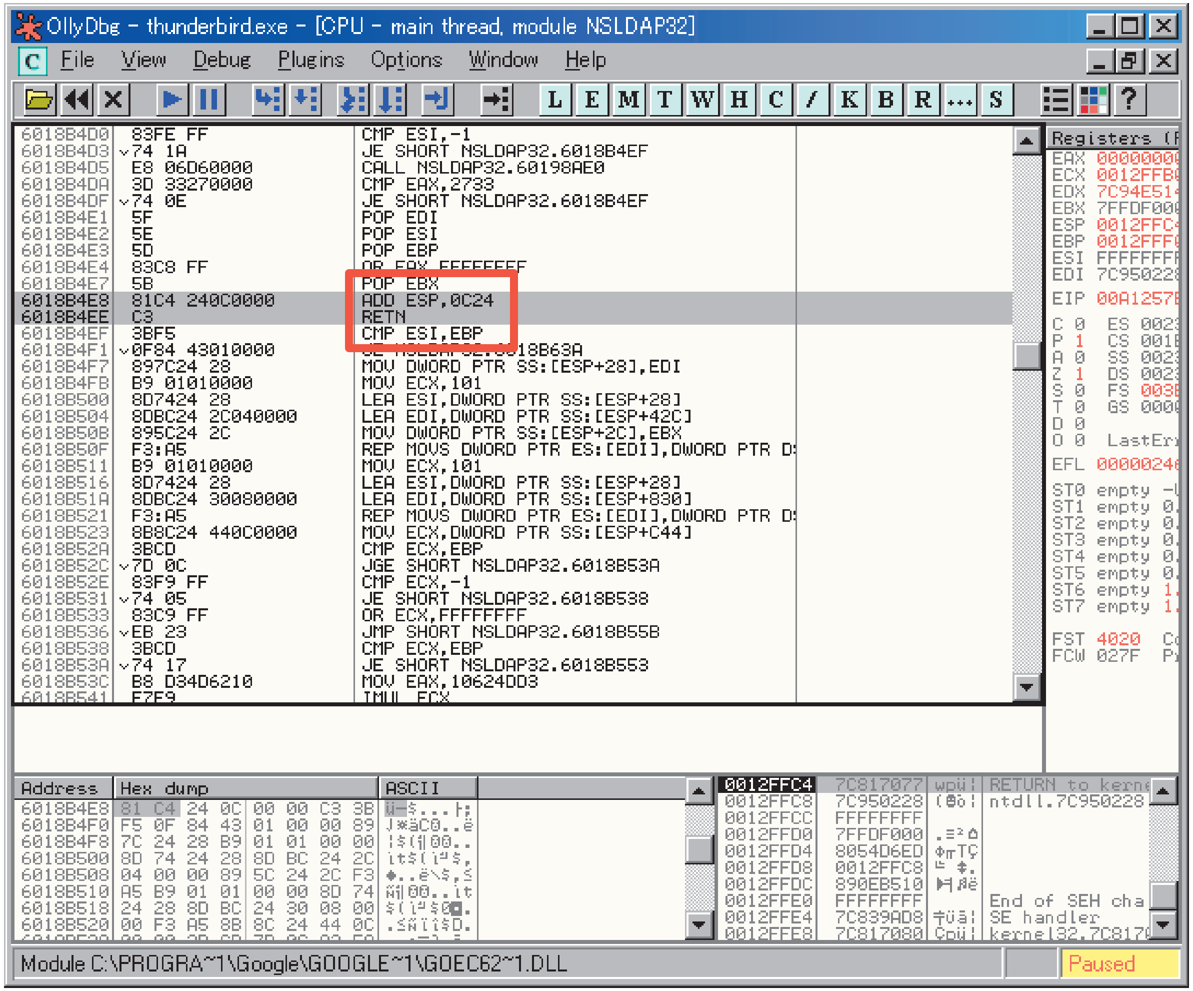
Module(EXE or DLL) No SafeSEH

add esp, 0x0C24
retn

Exception handlers which can be used



Example of “add esp + retn”


OllyDbg - thunderbird.exe - [CPU - main thread, module NSLDAP32]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Registers (F
Address Hex dump ASCII
Paused

```
6018B4D0   83FE FF          CMP ESI,-1
6018B4D3  v74 1A            JE SHORT NSLDAP32.6018B4EF
6018B4D5   E8 06D60000      CALL NSLDAP32.60198AE0
6018B4DA   3D 33270000      CMP EAX,2733
6018B4DF  v74 0E            JE SHORT NSLDAP32.6018B4EF
6018B4E1   5F               POP EDI
6018B4E2   5E               POP ESI
6018B4E3   5D               POP EBP
6018B4E4   83C8 FF          OR EAX,FFFFFFFF
6018B4E7   5B               POP EBX
6018B4E8   81C4 240C0000    ADD ESP,0C24
6018B4EE   C3               RETN
6018B4EF   3BF5             CMP ESI,EBP
6018B4F1  v0F84 43010000    JE NSLDAP32.6018B63A
6018B4F7   897C24 28        MOV DWORD PTR SS:[ESP+28],EDI
6018B4FB   B9 01010000      MOV ECX,101
6018B500   8D7424 28        LEA ESI,DWORD PTR SS:[ESP+28]
6018B504   8DBC24 2C040000  LEA EDI,DWORD PTR SS:[ESP+42C]
6018B50B   895C24 2C        MOV DWORD PTR SS:[ESP+2C],EBX
6018B50F   F3:A5            REP MOVS DWORD PTR ES:[EDI],DWORD PTR D:
6018B511   B9 01010000      MOV ECX,101
6018B516   8D7424 28        LEA ESI,DWORD PTR SS:[ESP+28]
6018B51A   8DBC24 30080000  LEA EDI,DWORD PTR SS:[ESP+830]
6018B521   F3:A5            REP MOVS DWORD PTR ES:[EDI],DWORD PTR D:
6018B523   8B8C24 440C0000  MOV ECX,DWORD PTR SS:[ESP+C44]
6018B52A   3BCD             CMP ECX,EBP
6018B52C  v7D 0C            JGE SHORT NSLDAP32.6018B53A
6018B52E   83F9 FF          CMP ECX,-1
6018B531  v74 05            JE SHORT NSLDAP32.6018B538
6018B533   83C9 FF          OR ECX,FFFFFFFF
6018B536  vEB 23            JMP SHORT NSLDAP32.6018B55B
6018B538   3BCD             CMP ECX,EBP
6018B53A  v74 17            JE SHORT NSLDAP32.6018B553
6018B53C   B8 D34D6210      MOV EAX,10624DD3
6018B541   F7E9             IMUL ECX
```

```
Registers (F
EAX 0000000
ECX 0012FFB
EDX 7C94E51
EBX 7FFDF00
ESP 0012FFC
EBP 0012FFF
ESI FFFFFFF
EDI 7C95022

EIP 00A1257

C 0  ES 002
P 1  CS 001
A 0  SS 002
Z 1  DS 002
S 0  FS 003
T 0  GS 000
D 0
O 0  LastEr

EFL 0000024

ST0 empty -
ST1 empty 0.
ST2 empty 0.
ST3 empty 0.
ST4 empty 0.
ST5 empty 0.
ST6 empty 1.
ST7 empty 1.

FST 4020  C
FCW 027F  P
```

```
Address  Hex dump
6018B4E8 81 C4 24 0C 00 00 C3 3B
6018B4F0 F5 0F 84 43 01 00 00 89
6018B4F8 7C 24 28 B9 01 01 00 00
6018B500 8D 74 24 28 8D BC 24 2C
6018B508 04 00 00 89 5C 24 2C F3
6018B510 A5 B9 01 01 00 00 8D 74
6018B518 24 28 8D BC 24 30 08 00
6018B520 00 F3 A5 8B 8C 24 44 0C
```

```
0012FFC4  7C817077  RETURN to kerne
0012FFC8  7C950228  ntdll.7C950228
0012FFCC  FFFFFFFF
0012FFD0  7FFDF000
0012FFD4  8054D6ED
0012FFD8  0012FFC8
0012FFDC  890EB510
0012FFE0  FFFFFFFF  End of SEH cha
0012FFE4  7C839AD8  SE handler
0012FFE8  7C817080  kernel32.7C817
```

Module C:\PROGRA~1\Google\GOOGLE~1\GOEC62~1.DLL

From Thunderbird 2.0.0.23

Summarize the condition

- The address of the stack where buffer overflow occurs is known.
- The address of the FinalExceptionHandler in ntdll.dll is known.
- A process has a module not protected by /SafeSEH
- “add esp + retn” instructions which matches followings can be found in the module.
  - The amount of “add esp” rewind is larger than the stack made by windows exception dispatcher.
  - The amount of “add esp” rewind is smaller than the stack size when the exception handler is called.
- The address of VirtualAlloc and memcpy is known (or some alternatives can be used).
- Attacker can write 0x00 on the stack by using buffer overflow, which means string functions can’t be used to exploit. (At least the method I showed here can not be used if this doesn’t match)

Demonstration

- Demonstration on Windows 7
- I assumed that “add esp,0x0c24 – retn” can be found in a non /SafeSEH protected module.
- /GS, Software DEP , Hardware DEP, SEHOP are all enabled.
- ASLR is disabled.

Importance of ASLR

- Makes it difficult to recreate proper SEH chain
- Makes it difficult to find “add esp + retn” instructions at a fixed address.
- Makes it difficult to set FinalExceptionHandler address in the last element of SEH chain.

Conclusion

Protections	Windows XP SP3	Windows Vista SP1	Windows 7
/GS + SafeSEH	Exploitable by using data area as an exception handler	Exploitable by using data area as an exception handler	Exploitable by using data area as an exception handler
/GS + SafeSEH + Software DEP	If all modules are SafeSEH protected its difficult to exploit	If all modules are SafeSEH protected its difficult to exploit	If all modules are SafeSEH protected its difficult to exploit
/GS + Software DEP + Hardware DEP	Exploitable by Return-into-libc or Return-oriented programming	Exploitable by Return-into-libc or Return-oriented programming	Exploitable by Return-into-libc or Return-oriented programming
/GS + Software DEP + SEHOP	-	Exploitable by recreating proper SEH chain	Exploitable by recreating proper SEH chain
/GS + SafeSEH + SEHOP	-	Exploitable by recreating proper SEH chain and using data area as an exception handler	Exploitable by recreating proper SEH chain and using data area as an exception handler
/GS + Software DEP + SEHOP + Hardware DEP	-	Exploitable by recreating proper SEH Chain and using data area and return-oriented programming	Exploitable by recreating proper SEH Chain and using data area and return-oriented programming
/GS + SEHOP + ASLR	-	Difficult to exploit	Difficult to exploit
/GS + Software DEP + SEHOP + Hardware DEP + ASLR	-	Difficult to exploit	Difficult to exploit

ASLR + DEP

- “Bypassing Browser Memory Protections” (Alexander Sotirov, Mark Dowd) shows how to bypass them.
- But it was without SEHOP

Questions?