```
text:00B11000 ; Section 2. (virtual address 00011000)
.text:00B11000 ; Virtual size                  : 0000739E (  29598.)
.text:00B11000 ; Section size in file          : 00007400 (  29696.)
.text:00B11000 ; Offset to raw data for section: 00000400
.text:00B11000 ; Flags 60000020: Text Executable Readable
.text:00B11000 ; Alignment     : default
.text:00B11000 ; ═══════════════════════════════════════════════════════════════════════════
.text:00B11000
.text:00B11000 ; Segment type: Pure code
.text:00B11000 ; Segment permissions: Read/Execute
.text:00B11000 _text           segment para public 'CODE' use32
.text:00B11000                 assume cs:_text
.text:00B11000                 ;org 0B11000h
.text:00B11000                 assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing
.text:00B11000                 db 5 dup(0CCh)
.text:00B11005 ; [00000005 BYTES: COLLAPSED FUNCTION j_terminate. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1100A
.text:00B1100A ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1100A
.text:00B1100A ; Attributes: thunk
.text:00B1100A
.text:00B1100A sub_B1100A      proc near               ; CODE XREF: sub_B127A0+2C↓p
.text:00B1100A                 jmp     sub_B147F0
.text:00B1100A sub_B1100A      endp
.text:00B1100A
.text:00B1100F
.text:00B1100F ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1100F
.text:00B1100F ; Attributes: thunk
.text:00B1100F
.text:00B1100F sub_B1100F      proc near               ; CODE XREF: sub_B127A0+D↓p
.text:00B1100F                 jmp     sub_B12C10
.text:00B1100F sub_B1100F      endp
.text:00B1100F
.text:00B11014 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11014                 jmp     loc_B15CC1
.text:00B11019 ; [00000005 BYTES: COLLAPSED FUNCTION j__register_onexit_function. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1101E
.text:00B1101E ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1101E
.text:00B1101E ; Attributes: thunk
.text:00B1101E
.text:00B1101E sub_B1101E      proc near               ; CODE XREF: sub_B13CF0+33↓p
.text:00B1101E                                         ; sub_B14010+2C↓p
.text:00B1101E                 jmp     sub_B15D20
.text:00B1101E sub_B1101E      endp
.text:00B1101E
.text:00B11023
.text:00B11023 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11023
.text:00B11023 ; Attributes: thunk
.text:00B11023
.text:00B11023 ; int start()
.text:00B11023                 public start
.text:00B11023 start           proc near
.text:00B11023                 jmp     start_0
.text:00B11023 start           endp
.text:00B11023
.text:00B11028 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11028                 jmp     loc_B15C7F
.text:00B1102D
.text:00B1102D ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1102D
.text:00B1102D ; Attributes: thunk
.text:00B1102D
.text:00B1102D sub_B1102D      proc near               ; CODE XREF: __security_check_cookie(x):loc_B12709↓j
.text:00B1102D                 jmp     sub_B134F0
.text:00B1102D sub_B1102D      endp
.text:00B1102D
.text:00B11032 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11032                 jmp     loc_B13910
.text:00B11037
.text:00B11037 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11037
.text:00B11037 ; Attributes: thunk
```

```
.text:00B11037
.text:00B11037 ; int sub_B11037(void)
.text:00B11037 sub_B11037      proc near               ; CODE XREF: sub_B128D0:loc_B129F1↓p
.text:00B11037                 jmp     sub_B14440
.text:00B11037 sub_B11037      endp
.text:00B11037
.text:00B1103C
.text:00B1103C ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1103C
.text:00B1103C ; Attributes: thunk
.text:00B1103C
.text:00B1103C sub_B1103C      proc near               ; CODE XREF: sub_B127A0+6E↓p
.text:00B1103C                 jmp     sub_B14390
.text:00B1103C sub_B1103C      endp
.text:00B1103C
.text:00B11041 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11041                 jmp     loc_B15CA3
.text:00B11041 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11046                 dw 55E9h
.text:00B11048                 dd 0E9000013h, 12D0h, 25EBE9h, 1430E900h
.text:00B11058                 db 2 dup(0)
.text:00B1105A
.text:00B1105A ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1105A
.text:00B1105A ; Attributes: thunk
.text:00B1105A
.text:00B1105A ; int sub_B1105A(void)
.text:00B1105A sub_B1105A      proc near               ; CODE XREF: sub_B128D0+DE↓p
.text:00B1105A                 jmp     sub_B14430
.text:00B1105A sub_B1105A      endp
.text:00B1105A
.text:00B1105F ; ───────────────────────────────────────────────────────────────────────────
.text:00B1105F                 jmp     loc_B14C60
.text:00B11064
.text:00B11064 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11064
.text:00B11064 ; Attributes: thunk
.text:00B11064
.text:00B11064 sub_B11064      proc near               ; CODE XREF: sub_B14630+3↓p
.text:00B11064                 jmp     sub_B14720
.text:00B11064 sub_B11064      endp
.text:00B11064
.text:00B11069 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11069                 jmp     loc_B12478
.text:00B11069 ; ───────────────────────────────────────────────────────────────────────────
.text:00B1106E                 dw 5DE9h
.text:00B11070                 db 11h, 2 dup(0)
.text:00B11073 ; [00000005 BYTES: COLLAPSED FUNCTION j__crt_at_quick_exit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11078 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11078                 jmp     loc_B13CD0
.text:00B1107D ; [00000005 BYTES: COLLAPSED FUNCTION j___p___argc. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11082 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11082                 jmp     sub_B15D40
.text:00B11087 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11087                 jmp     loc_B14970
.text:00B1108C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1108C                 jmp     loc_B1247E
.text:00B11091 ; [00000005 BYTES: COLLAPSED FUNCTION j__initterm_e. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11096
.text:00B11096 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11096
.text:00B11096 ; Attributes: thunk
.text:00B11096
.text:00B11096 sub_B11096      proc near               ; CODE XREF: sub_B13030+110↓p
.text:00B11096                 jmp     sub_B15240
.text:00B11096 sub_B11096      endp
.text:00B11096
.text:00B1109B ; [00000005 BYTES: COLLAPSED FUNCTION j__initialize_narrow_environment. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B110A0
.text:00B110A0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B110A0
.text:00B110A0 ; Attributes: thunk
.text:00B110A0
.text:00B110A0 sub_B110A0      proc near               ; CODE XREF: sub_B13030+45↓p
```

```
.text:00B110A0                 jmp     sub_B138D0
.text:00B110A0 sub_B110A0      endp
.text:00B110A0
.text:00B110A5 ; [00000005 BYTES: COLLAPSED FUNCTION j__initialize_onexit_table. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B110AA
.text:00B110AA ; =============== S U B R O U T I N E =======================================
.text:00B110AA
.text:00B110AA ; Attributes: thunk
.text:00B110AA
.text:00B110AA sub_B110AA      proc near               ; CODE XREF: sub_B127A0:loc_B127EE↓p
.text:00B110AA                 jmp     sub_B14300
.text:00B110AA sub_B110AA      endp
.text:00B110AA
.text:00B110AF ; [00000005 BYTES: COLLAPSED FUNCTION j___stdio_common_vsprintf_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B110B4 ; ---------------------------------------------------------------------------
.text:00B110B4                 jmp     loc_B14BC0
.text:00B110B9
.text:00B110B9 ; =============== S U B R O U T I N E =======================================
.text:00B110B9
.text:00B110B9 ; Attributes: thunk
.text:00B110B9
.text:00B110B9 sub_B110B9      proc near               ; CODE XREF: .text:00B14C69↓p
.text:00B110B9                 jmp     sub_B148F0
.text:00B110B9 sub_B110B9      endp
.text:00B110B9
.text:00B110BE ; [00000005 BYTES: COLLAPSED FUNCTION j__execute_onexit_table. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B110C3 ; ---------------------------------------------------------------------------
.text:00B110C3                 jmp     sub_B15D20
.text:00B110C8 ; ---------------------------------------------------------------------------
.text:00B110C8                 jmp     loc_B12472
.text:00B110CD
.text:00B110CD ; =============== S U B R O U T I N E =======================================
.text:00B110CD
.text:00B110CD ; Attributes: thunk
.text:00B110CD
.text:00B110CD sub_B110CD      proc near               ; CODE XREF: sub_B127A0+3↓p
.text:00B110CD                 jmp     sub_B12BF0
.text:00B110CD sub_B110CD      endp
.text:00B110CD
.text:00B110D2 ; [00000005 BYTES: COLLAPSED FUNCTION j__wsplitpath_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B110D7 ; ---------------------------------------------------------------------------
.text:00B110D7                 jmp     loc_B13BA0
.text:00B110DC ; ---------------------------------------------------------------------------
.text:00B110DC                 jmp     loc_B1246C
.text:00B110E1 ; ---------------------------------------------------------------------------
.text:00B110E1                 jmp     loc_B145E0
.text:00B110E6 ; [00000005 BYTES: COLLAPSED FUNCTION j___current_exception. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B110EB
.text:00B110EB ; =============== S U B R O U T I N E =======================================
.text:00B110EB
.text:00B110EB ; Attributes: thunk
.text:00B110EB
.text:00B110EB sub_B110EB      proc near               ; CODE XREF: .text:00B1275A↓p
.text:00B110EB                                         ; sub_B12780+A↓p
.text:00B110EB                 jmp     sub_B12730
.text:00B110EB sub_B110EB      endp
.text:00B110EB
.text:00B110F0 ; ---------------------------------------------------------------------------
.text:00B110F0                 jmp     loc_B15CCD
.text:00B110F5 ; ---------------------------------------------------------------------------
.text:00B110F5                 jmp     loc_B15C5B
.text:00B110FA
.text:00B110FA ; =============== S U B R O U T I N E =======================================
.text:00B110FA
.text:00B110FA ; Attributes: thunk
.text:00B110FA
.text:00B110FA sub_B110FA      proc near               ; CODE XREF: sub_B12C30+3↓p
.text:00B110FA                 jmp     sub_B142D0
.text:00B110FA sub_B110FA      endp
.text:00B110FA
.text:00B110FF ; [00000005 BYTES: COLLAPSED FUNCTION j__CrtDbgReportW. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11104
.text:00B11104 ; =============== S U B R O U T I N E =======================================
.text:00B11104
.text:00B11104 ; Attributes: thunk
```

```
.text:00B11104
.text:00B11104 ; int __cdecl sub_B11104(int, int)
.text:00B11104 sub_B11104      proc near               ; CODE XREF: sub_B12680+14↓p
.text:00B11104                 jmp     sub_B12E30
.text:00B11104 sub_B11104      endp
.text:00B11104
.text:00B11109 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11109                 jmp     loc_B15C9D
.text:00B1110E ; ───────────────────────────────────────────────────────────────────────────
.text:00B1110E                 jmp     loc_B15CD9
.text:00B1110E ; ───────────────────────────────────────────────────────────────────────────
.text:00B11113                 db 0E9h
.text:00B11114                 dd 11B8h
.text:00B11118 ; [00000005 BYTES: COLLAPSED FUNCTION j__set_app_type. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1111D ; ───────────────────────────────────────────────────────────────────────────
.text:00B1111D                 jmp     loc_B13C30
.text:00B11122
.text:00B11122 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11122
.text:00B11122 ; Attributes: thunk
.text:00B11122
.text:00B11122 ; int __cdecl UserMathErrorFunction(struct _exception *)
.text:00B11122 UserMathErrorFunction proc near         ; DATA XREF: sub_B127A0+5C↓o
.text:00B11122                 jmp     sub_B142A0
.text:00B11122 UserMathErrorFunction endp
.text:00B11122
.text:00B11127 ; [00000005 BYTES: COLLAPSED FUNCTION j__register_thread_local_exe_atexit_callback. PRESS
CTRL-NUMPAD+ TO EXPAND]
.text:00B1112C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1112C                 jmp     loc_B13C90
.text:00B11131 ; [00000005 BYTES: COLLAPSED FUNCTION j__crt_atexit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11136
.text:00B11136 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11136
.text:00B11136 ; Attributes: thunk
.text:00B11136
.text:00B11136 sub_B11136      proc near               ; CODE XREF: .text:00B13BEF↓p
.text:00B11136                                         ; .text:00B13C18↓p
.text:00B11136                 jmp     sub_B15D10
.text:00B11136 sub_B11136      endp
.text:00B11136
.text:00B1113B ; [00000005 BYTES: COLLAPSED FUNCTION j__seh_filter_exe. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11140 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11140                 jmp     loc_B15CB5
.text:00B11145 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11145                 jmp     loc_B149E0
.text:00B1114A ; ───────────────────────────────────────────────────────────────────────────
.text:00B1114A                 jmp     loc_B12490
.text:00B1114F
.text:00B1114F ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1114F
.text:00B1114F ; Attributes: thunk
.text:00B1114F
.text:00B1114F sub_B1114F      proc near               ; CODE XREF: sub_B12C10+4↓p
.text:00B1114F                 jmp     sub_B142C0
.text:00B1114F sub_B1114F      endp
.text:00B1114F
.text:00B11154
.text:00B11154 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11154
.text:00B11154 ; Attributes: thunk
.text:00B11154
.text:00B11154 sub_B11154      proc near               ; CODE XREF: sub_B128D0+F2↓p
.text:00B11154                                         ; sub_B128D0+135↓p
.text:00B11154                 jmp     sub_B13E60
.text:00B11154 sub_B11154      endp
.text:00B11154
.text:00B11159 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11159                 jmp     loc_B11960
.text:00B1115E
.text:00B1115E ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1115E
.text:00B1115E ; Attributes: thunk
.text:00B1115E
.text:00B1115E sub_B1115E      proc near               ; CODE XREF: sub_B14010+1F↓p
```

```
.text:00B1115E                 jmp     sub_B15D20
.text:00B1115E sub_B1115E      endp
.text:00B1115E
.text:00B11163
.text:00B11163 ; =============== S U B R O U T I N E =======================================
.text:00B11163
.text:00B11163 ; Attributes: thunk
.text:00B11163
.text:00B11163 sub_B11163      proc near               ; CODE XREF: sub_B127A0+92↓p
.text:00B11163                                         ; .text:loc_B13B80↓p
.text:00B11163                 jmp     sub_B12B90
.text:00B11163 sub_B11163      endp
.text:00B11163
.text:00B11168
.text:00B11168 ; =============== S U B R O U T I N E =======================================
.text:00B11168
.text:00B11168 ; Attributes: thunk
.text:00B11168
.text:00B11168 sub_B11168      proc near               ; CODE XREF: sub_B127A0+14↓p
.text:00B11168                                         ; .text:00B13BA5↓p
.text:00B11168                 jmp     sub_B13D50
.text:00B11168 sub_B11168      endp
.text:00B11168
.text:00B1116D ; [00000005 BYTES: COLLAPSED FUNCTION j___p__commode. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11172 ; ---------------------------------------------------------------------------
.text:00B11172                 jmp     loc_B13B60
.text:00B11177 ; ---------------------------------------------------------------------------
.text:00B11177                 jmp     loc_B124B4
.text:00B1117C
.text:00B1117C ; =============== S U B R O U T I N E =======================================
.text:00B1117C
.text:00B1117C ; Attributes: thunk
.text:00B1117C
.text:00B1117C sub_B1117C      proc near               ; CODE XREF: sub_B12890+8↓p
.text:00B1117C                 jmp     sub_B142E0
.text:00B1117C sub_B1117C      endp
.text:00B1117C
.text:00B11181 ; [00000005 BYTES: COLLAPSED FUNCTION __security_check_cookie(x). PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11186 ; [00000005 BYTES: COLLAPSED FUNCTION j_memset. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1118B ; ---------------------------------------------------------------------------
.text:00B1118B                 jmp     loc_B138F0
.text:00B11190 ; ---------------------------------------------------------------------------
.text:00B11190                 jmp     loc_B13980
.text:00B11195 ; [00000005 BYTES: COLLAPSED FUNCTION j___vcrt_GetModuleHandleW. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1119A ; ---------------------------------------------------------------------------
.text:00B1119A                 jmp     loc_B12720
.text:00B1119F ; [00000005 BYTES: COLLAPSED FUNCTION j__seh_filter_dll. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B111A4
.text:00B111A4 ; =============== S U B R O U T I N E =======================================
.text:00B111A4
.text:00B111A4 ; Attributes: thunk
.text:00B111A4
.text:00B111A4 sub_B111A4      proc near               ; CODE XREF: sub_B12880+3↓p
.text:00B111A4                 jmp     sub_B143B0
.text:00B111A4 sub_B111A4      endp
.text:00B111A4
.text:00B111A4 ; ---------------------------------------------------------------------------
.text:00B111A9                 db 0E9h, 0C2h, 6
.text:00B111AC                 db 2 dup(0)
.text:00B111AE
.text:00B111AE ; =============== S U B R O U T I N E =======================================
.text:00B111AE
.text:00B111AE ; Attributes: thunk
.text:00B111AE
.text:00B111AE sub_B111AE      proc near               ; CODE XREF: sub_B128D0+D6↓p
.text:00B111AE                 jmp     sub_B13FE0
.text:00B111AE sub_B111AE      endp
.text:00B111AE
.text:00B111B3 ; ---------------------------------------------------------------------------
.text:00B111B3                 jmp     loc_B15C61
.text:00B111B8 ; ---------------------------------------------------------------------------
.text:00B111B8                 jmp     loc_B11E00
.text:00B111BD ; ---------------------------------------------------------------------------
.text:00B111BD                 jmp     loc_B15C97
.text:00B111C2
```

```
.text:00B111C2 ; =============== S U B R O U T I N E =======================================
.text:00B111C2
.text:00B111C2 ; Attributes: thunk
.text:00B111C2
.text:00B111C2 sub_B111C2      proc near               ; CODE XREF: sub_B128D0+180↓p
.text:00B111C2                 jmp     sub_B14010
.text:00B111C2 sub_B111C2      endp
.text:00B111C2
.text:00B111C7
.text:00B111C7 ; =============== S U B R O U T I N E =======================================
.text:00B111C7
.text:00B111C7 ; Attributes: thunk
.text:00B111C7
.text:00B111C7 sub_B111C7      proc near               ; CODE XREF: .text:00B125A1↓p
.text:00B111C7                                         ; .text:00B125BC↓p
.text:00B111C7                 jmp     sub_B12D10
.text:00B111C7 sub_B111C7      endp
.text:00B111C7
.text:00B111CC ; ---------------------------------------------------------------------------
.text:00B111CC                 jmp     loc_B15C6D
.text:00B111D1
.text:00B111D1 ; =============== S U B R O U T I N E =======================================
.text:00B111D1
.text:00B111D1 ; Attributes: thunk
.text:00B111D1
.text:00B111D1 sub_B111D1      proc near               ; CODE XREF: .text:00B13C13↓p
.text:00B111D1                 jmp     sub_B15D10
.text:00B111D1 sub_B111D1      endp
.text:00B111D1
.text:00B111D6 ; ---------------------------------------------------------------------------
.text:00B111D6                 jmp     loc_B13920
.text:00B111DB
.text:00B111DB ; =============== S U B R O U T I N E =======================================
.text:00B111DB
.text:00B111DB ; Attributes: thunk
.text:00B111DB
.text:00B111DB ; void __cdecl Function()
.text:00B111DB Function        proc near               ; DATA XREF: sub_B127A0+31↓o
.text:00B111DB                 jmp     sub_B14830
.text:00B111DB Function        endp
.text:00B111DB
.text:00B111E0 ; ---------------------------------------------------------------------------
.text:00B111E0                 jmp     loc_B15CDF
.text:00B111E5
.text:00B111E5 ; =============== S U B R O U T I N E =======================================
.text:00B111E5
.text:00B111E5 ; Attributes: thunk
.text:00B111E5
.text:00B111E5 sub_B111E5      proc near               ; CODE XREF: sub_B13B00+6↓p
.text:00B111E5                                         ; .text:00B13B63↓p ...
.text:00B111E5                 jmp     sub_B15B20
.text:00B111E5 sub_B111E5      endp
.text:00B111E5
.text:00B111EA
.text:00B111EA ; =============== S U B R O U T I N E =======================================
.text:00B111EA
.text:00B111EA ; Attributes: thunk
.text:00B111EA
.text:00B111EA sub_B111EA      proc near               ; CODE XREF: .text:00B13B6C↓p
.text:00B111EA                                         ; sub_B13CF0:loc_B13D00↓p
.text:00B111EA                 jmp     sub_B15720
.text:00B111EA sub_B111EA      endp
.text:00B111EA
.text:00B111EF ; ---------------------------------------------------------------------------
.text:00B111EF                 jmp     sub_B15D30
.text:00B111F4 ; ---------------------------------------------------------------------------
.text:00B111F4                 jmp     loc_B13760
.text:00B111F9 ; ---------------------------------------------------------------------------
.text:00B111F9                 jmp     loc_B15CD3
.text:00B111FE ; [00000005 BYTES: COLLAPSED FUNCTION j_wcscpy_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11203 ; ---------------------------------------------------------------------------
.text:00B11203                 jmp     loc_B124A8
.text:00B11208 ; ---------------------------------------------------------------------------
.text:00B11208                 jmp     loc_B12120
.text:00B1120D
```

```
.text:00B1120D ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1120D
.text:00B1120D ; Attributes: thunk
.text:00B1120D
.text:00B1120D ; int __cdecl sub_B1120D(char *Buffer, size_t BufferCount, char *Format, _locale_t Locale, va_list
ArgList)
.text:00B1120D sub_B1120D      proc near               ; CODE XREF: sub_B13480+12↓p
.text:00B1120D
.text:00B1120D Buffer          = dword ptr  4
.text:00B1120D BufferCount     = dword ptr  8
.text:00B1120D Format          = dword ptr  0Ch
.text:00B1120D Locale          = dword ptr  10h
.text:00B1120D ArgList         = dword ptr  14h
.text:00B1120D
.text:00B1120D                 jmp     sub_B13440
.text:00B1120D sub_B1120D      endp
.text:00B1120D
.text:00B11212 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11212                 jmp     loc_B15C91
.text:00B11217 ; [00000005 BYTES: COLLAPSED FUNCTION j__set_new_mode. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1121C
.text:00B1121C ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1121C
.text:00B1121C ; Attributes: thunk
.text:00B1121C
.text:00B1121C sub_B1121C      proc near               ; CODE XREF: sub_B127A0+25↓p
.text:00B1121C                                         ; sub_B127A0+49↓p ...
.text:00B1121C                 jmp     sub_B14450
.text:00B1121C sub_B1121C      endp
.text:00B1121C
.text:00B11221 ; [00000005 BYTES: COLLAPSED FUNCTION j___vcrt_LoadLibraryExW. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11226 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11226                 jmp     loc_B15C79
.text:00B1122B
.text:00B1122B ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1122B
.text:00B1122B ; Attributes: thunk
.text:00B1122B
.text:00B1122B ; int __fastcall CGC_2(int, int, int, int)
.text:00B1122B CGC_2           proc near               ; CODE XREF: _main_0+5A↓p
.text:00B1122B                 jmp     sub_B12610
.text:00B1122B CGC_2           endp
.text:00B1122B
.text:00B11230 ; [00000005 BYTES: COLLAPSED FUNCTION j__except_handler4_common. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11235
.text:00B11235 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11235
.text:00B11235 ; Attributes: thunk
.text:00B11235
.text:00B11235 sub_B11235      proc near               ; CODE XREF: .text:loc_B13CAB↓p
.text:00B11235                 jmp     sub_B15D40
.text:00B11235 sub_B11235      endp
.text:00B11235
.text:00B1123A
.text:00B1123A ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1123A
.text:00B1123A ; Attributes: thunk
.text:00B1123A
.text:00B1123A ; int __cdecl sub_B1123A(_PVFV Function)
.text:00B1123A sub_B1123A      proc near               ; CODE XREF: sub_B127A0+36↓p
.text:00B1123A
.text:00B1123A Function        = dword ptr  4
.text:00B1123A
.text:00B1123A                 jmp     sub_B14130
.text:00B1123A sub_B1123A      endp
.text:00B1123A
.text:00B1123F ; ───────────────────────────────────────────────────────────────────────────
.text:00B1123F                 jmp     loc_B15C8B
.text:00B11244 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11244                 jmp     loc_B14940
.text:00B11249 ; [00000005 BYTES: COLLAPSED FUNCTION j_exit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1124E
.text:00B1124E ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1124E
.text:00B1124E ; Attributes: thunk
```

```
.text:00B1124E
.text:00B1124E sub_B1124E      proc near               ; CODE XREF: .text:00B14AAD↓p
.text:00B1124E                 jmp     sub_B14910
.text:00B1124E sub_B1124E      endp
.text:00B1124E
.text:00B11253 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11253                 jmp     A_func1
.text:00B11258
.text:00B11258 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11258
.text:00B11258 ; Attributes: thunk
.text:00B11258
.text:00B11258 sub_B11258      proc near               ; CODE XREF: sub_B127A0+3E↓p
.text:00B11258                                         ; .text:loc_B13B73↓p
.text:00B11258                 jmp     sub_B12B70
.text:00B11258 sub_B11258      endp
.text:00B11258
.text:00B1125D
.text:00B1125D ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1125D
.text:00B1125D ; Attributes: thunk
.text:00B1125D
.text:00B1125D sub_B1125D      proc near               ; CODE XREF: sub_B127A0+53↓p
.text:00B1125D                 jmp     sub_B14400
.text:00B1125D sub_B1125D      endp
.text:00B1125D
.text:00B11262
.text:00B11262 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11262
.text:00B11262 ; Attributes: thunk
.text:00B11262
.text:00B11262 ; int __cdecl sub_B11262(_PVFV Function)
.text:00B11262 sub_B11262      proc near               ; CODE XREF: sub_B14130+8↓p
.text:00B11262
.text:00B11262 Function        = dword ptr  4
.text:00B11262
.text:00B11262                 jmp     sub_B14060
.text:00B11262 sub_B11262      endp
.text:00B11262
.text:00B11267
.text:00B11267 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11267
.text:00B11267 ; Attributes: thunk
.text:00B11267
.text:00B11267 sub_B11267      proc near               ; CODE XREF: sub_B127A0+9C↓p
.text:00B11267                 jmp     sub_B14630
.text:00B11267 sub_B11267      endp
.text:00B11267
.text:00B1126C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1126C                 jmp     loc_B140E0
.text:00B11271 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11271                 jmp     sub_B14720
.text:00B11276 ; [00000005 BYTES: COLLAPSED FUNCTION j__c_exit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1127B ; [00000005 BYTES: COLLAPSED FUNCTION j__wmakepath_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11280 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11280                 jmp     loc_B13BD0
.text:00B11285 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11285                 jmp     loc_B15C73
.text:00B1128A ; ───────────────────────────────────────────────────────────────────────────
.text:00B1128A                 jmp     sub_B15D50
.text:00B1128F ; ───────────────────────────────────────────────────────────────────────────
.text:00B1128F
.text:00B1128F loc_B1128F:                             ; DATA XREF: .rdata:00B1ACDC↓o
.text:00B1128F                 jmp     loc_B12740
.text:00B11294
.text:00B11294 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11294
.text:00B11294 ; Attributes: thunk
.text:00B11294
.text:00B11294 ; int CGC_1(void)
.text:00B11294 CGC_1           proc near               ; CODE XREF: .text:00B1198C↓p
.text:00B11294                                         ; sub_B119A0+4A↓p ...
.text:00B11294                 jmp     sub_B12680
.text:00B11294 CGC_1           endp
.text:00B11294
```

```
.text:00B11299
.text:00B11299 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11299
.text:00B11299 ; Attributes: thunk
.text:00B11299
.text:00B11299 sub_B11299      proc near               ; CODE XREF: .text:00B13BD3↓p
.text:00B11299                 jmp     sub_B15D00
.text:00B11299 sub_B11299      endp
.text:00B11299
.text:00B1129E ; [00000005 BYTES: COLLAPSED FUNCTION j__configthreadlocale. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B112A3 ; ───────────────────────────────────────────────────────────────────────────
.text:00B112A3                 jmp     loc_B1249C
.text:00B112A8
.text:00B112A8 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B112A8
.text:00B112A8 ; Attributes: thunk
.text:00B112A8
.text:00B112A8 sub_B112A8      proc near               ; CODE XREF: .text:00B13CDD↓p
.text:00B112A8                 jmp     sub_B15D30
.text:00B112A8 sub_B112A8      endp
.text:00B112A8
.text:00B112AD ; ───────────────────────────────────────────────────────────────────────────
.text:00B112AD                 jmp     loc_B12484
.text:00B112B2 ; ───────────────────────────────────────────────────────────────────────────
.text:00B112B2                 jmp     loc_B14AA0
.text:00B112B7 ; ───────────────────────────────────────────────────────────────────────────
.text:00B112B7                 jmp     sub_B15CF0
.text:00B112BC ; ───────────────────────────────────────────────────────────────────────────
.text:00B112BC                 jmp     loc_B14320
.text:00B112C1 ; ───────────────────────────────────────────────────────────────────────────
.text:00B112C1                 jmp     loc_B13C10
.text:00B112C6
.text:00B112C6 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B112C6
.text:00B112C6 ; Attributes: thunk
.text:00B112C6
.text:00B112C6 sub_B112C6      proc near               ; CODE XREF: .text:loc_B13BE3↓p
.text:00B112C6                 jmp     sub_B15D00
.text:00B112C6 sub_B112C6      endp
.text:00B112C6
.text:00B112CB
.text:00B112CB ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B112CB
.text:00B112CB ; Attributes: thunk
.text:00B112CB
.text:00B112CB sub_B112CB      proc near               ; CODE XREF: sub_B128D0+118↓p
.text:00B112CB                                         ; sub_B13030+141↓p ...
.text:00B112CB                 jmp     sub_B148C0
.text:00B112CB sub_B112CB      endp
.text:00B112CB
.text:00B112D0
.text:00B112D0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B112D0
.text:00B112D0 ; Attributes: thunk
.text:00B112D0
.text:00B112D0 sub_B112D0      proc near               ; CODE XREF: sub_B127A0+78↓p
.text:00B112D0                 jmp     sub_B142F0
.text:00B112D0 sub_B112D0      endp
.text:00B112D0
.text:00B112D5
.text:00B112D5 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B112D5
.text:00B112D5 ; Attributes: thunk
.text:00B112D5
.text:00B112D5 sub_B112D5      proc near               ; CODE XREF: sub_B127A0:loc_B12837↓p
.text:00B112D5                 jmp     sub_B14640
.text:00B112D5 sub_B112D5      endp
.text:00B112D5
.text:00B112DA
.text:00B112DA ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B112DA
.text:00B112DA ; Attributes: thunk
.text:00B112DA
.text:00B112DA sub_B112DA      proc near               ; CODE XREF: sub_B13CF0+15↓p
.text:00B112DA                 jmp     sub_B15CF0
```

```
.text:00B112DA sub_B112DA      endp
.text:00B112DA
.text:00B112DF
.text:00B112DF ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B112DF
.text:00B112DF ; Attributes: thunk
.text:00B112DF
.text:00B112DF sub_B112DF      proc near               ; CODE XREF: sub_B13CF0:loc_B13D15↓p
.text:00B112DF                 jmp     sub_B15CF0
.text:00B112DF sub_B112DF      endp
.text:00B112DF
.text:00B112E4 ; ───────────────────────────────────────────────────────────────────────────
.text:00B112E4                 jmp     loc_B14B60
.text:00B112E9 ; ───────────────────────────────────────────────────────────────────────────
.text:00B112E9                 jmp     loc_B124C0
.text:00B112EE
.text:00B112EE ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B112EE
.text:00B112EE ; Attributes: thunk
.text:00B112EE
.text:00B112EE sub_B112EE      proc near               ; CODE XREF: sub_B12B70+3↓p
.text:00B112EE                 jmp     sub_B142B0
.text:00B112EE sub_B112EE      endp
.text:00B112EE
.text:00B112F3 ; ───────────────────────────────────────────────────────────────────────────
.text:00B112F3                 jmp     loc_B124A2
.text:00B112F8 ; ───────────────────────────────────────────────────────────────────────────
.text:00B112F8                 jmp     loc_B14920
.text:00B112FD ; [00000005 BYTES: COLLAPSED FUNCTION j__get_initial_narrow_environment. PRESS CTRL-NUMPAD+ TO
EXPAND]
.text:00B11302
.text:00B11302 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11302
.text:00B11302 ; Attributes: thunk
.text:00B11302
.text:00B11302 sub_B11302      proc near               ; CODE XREF: sub_B14450+1C↓p
.text:00B11302                                         ; sub_B14450+12A↓p
.text:00B11302                 jmp     sub_B147D0
.text:00B11302 sub_B11302      endp
.text:00B11302
.text:00B11307 ; [00000005 BYTES: COLLAPSED FUNCTION j___p___argv. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1130C
.text:00B1130C ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1130C
.text:00B1130C ; Attributes: thunk
.text:00B1130C
.text:00B1130C sub_B1130C      proc near               ; CODE XREF: sub_B128D0+35↓p
.text:00B1130C                 jmp     sub_B13CF0
.text:00B1130C sub_B1130C      endp
.text:00B1130C
.text:00B11311 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11311                 jmp     loc_B14AD0
.text:00B11316 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11316                 jmp     loc_B14980
.text:00B1131B ; ───────────────────────────────────────────────────────────────────────────
.text:00B1131B                 jmp     loc_B15CBB
.text:00B11320 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11320                 jmp     loc_B14B10
.text:00B11325 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11325                 jmp     loc_B124AE
.text:00B1132A ; [00000005 BYTES: COLLAPSED FUNCTION j___setusermatherr. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1132F ; ───────────────────────────────────────────────────────────────────────────
.text:00B1132F                 jmp     sub_B15D00
.text:00B11334
.text:00B11334 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11334
.text:00B11334 ; Attributes: thunk
.text:00B11334
.text:00B11334 ; int __cdecl sub_B11334(char *Buffer, size_t BufferCount, char *Format, char ArgList)
.text:00B11334 sub_B11334      proc near               ; CODE XREF: sub_B12D10+B5↓p
.text:00B11334                                         ; sub_B12F80+44↓p
.text:00B11334
.text:00B11334 Buffer          = dword ptr  4
.text:00B11334 BufferCount     = dword ptr  8
.text:00B11334 Format          = dword ptr  0Ch
```

```
.text:00B11334 ArgList         = byte ptr  10h
.text:00B11334
.text:00B11334                 jmp     sub_B13480
.text:00B11334 sub_B11334      endp
.text:00B11334
.text:00B11339 ; [00000005 BYTES: COLLAPSED FUNCTION _main. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1133E
.text:00B1133E ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1133E
.text:00B1133E ; Attributes: thunk
.text:00B1133E
.text:00B1133E sub_B1133E      proc near               ; CODE XREF: sub_B127A0+8↓p
.text:00B1133E                 jmp     sub_B12C30
.text:00B1133E sub_B1133E      endp
.text:00B1133E
.text:00B11343
.text:00B11343 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11343
.text:00B11343 ; Attributes: thunk
.text:00B11343
.text:00B11343 ; int sub_B11343(void)
.text:00B11343 sub_B11343      proc near               ; CODE XREF: sub_B128D0+56↓p
.text:00B11343                 jmp     sub_B13B00
.text:00B11343 sub_B11343      endp
.text:00B11343
.text:00B11348 ; [00000005 BYTES: COLLAPSED FUNCTION j___std_type_info_destroy_list. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1134D
.text:00B1134D ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1134D
.text:00B1134D ; Attributes: thunk
.text:00B1134D
.text:00B1134D sub_B1134D      proc near               ; CODE XREF: sub_B127A0+86↓p
.text:00B1134D                 jmp     sub_B14340
.text:00B1134D sub_B1134D      endp
.text:00B1134D
.text:00B11352
.text:00B11352 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11352
.text:00B11352 ; Attributes: thunk
.text:00B11352
.text:00B11352 sub_B11352      proc near               ; DATA XREF: .rdata:00B1AFE8↓o
.text:00B11352                 jmp     sub_B12780
.text:00B11352 sub_B11352      endp
.text:00B11352
.text:00B11357
.text:00B11357 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11357
.text:00B11357 ; Attributes: thunk
.text:00B11357
.text:00B11357 ; int sub_B11357(void)
.text:00B11357 sub_B11357      proc near               ; CODE XREF: sub_B128D0+15A↓p
.text:00B11357                                         ; sub_B128D0+1C7↓p
.text:00B11357                 jmp     sub_B14650
.text:00B11357 sub_B11357      endp
.text:00B11357
.text:00B1135C
.text:00B1135C ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1135C
.text:00B1135C ; Attributes: thunk
.text:00B1135C
.text:00B1135C sub_B1135C      proc near               ; CODE XREF: sub_B119A0+33↓p
.text:00B1135C                                         ; sub_B13440+3↓p ...
.text:00B1135C                 jmp     sub_B13430
.text:00B1135C sub_B1135C      endp
.text:00B1135C
.text:00B11361 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11361                 jmp     loc_B15C67
.text:00B11366 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11366                 jmp     loc_B13350
.text:00B1136B ; [00000005 BYTES: COLLAPSED FUNCTION j__controlfp_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11370 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11370                 jmp     sub_B15D10
.text:00B11375 ; [00000005 BYTES: COLLAPSED FUNCTION TopLevelExceptionFilter. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1137A
.text:00B1137A ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
```

```
.text:00B1137A
.text:00B1137A ; Attributes: thunk
.text:00B1137A
.text:00B1137A sub_B1137A      proc near               ; CODE XREF: sub_B148F0+7↓p
.text:00B1137A                 jmp     sub_B148D0
.text:00B1137A sub_B1137A      endp
.text:00B1137A
.text:00B1137F ; ───────────────────────────────────────────────────────────────────────────
.text:00B1137F                 jmp     loc_B15C85
.text:00B11384
.text:00B11384 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11384
.text:00B11384 ; Attributes: thunk
.text:00B11384
.text:00B11384 ; int __thiscall GetCurrentThreadId(void *this)
.text:00B11384 _GetCurrentThreadId proc near           ; CODE XREF: .text:00B1197C↓p
.text:00B11384                                         ; sub_B119A0+1C↓p ...
.text:00B11384                 jmp     sub_B126B0
.text:00B11384 _GetCurrentThreadId endp
.text:00B11384
.text:00B11389
.text:00B11389 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11389
.text:00B11389 ; Attributes: thunk
.text:00B11389
.text:00B11389 sub_B11389      proc near               ; CODE XREF: sub_B127A0+73↓p
.text:00B11389                 jmp     sub_B14350
.text:00B11389 sub_B11389      endp
.text:00B11389
.text:00B1138E
.text:00B1138E ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B1138E
.text:00B1138E ; Attributes: thunk
.text:00B1138E
.text:00B1138E sub_B1138E      proc near               ; CODE XREF: sub_B12890+3↓p
.text:00B1138E                 jmp     sub_B14700
.text:00B1138E sub_B1138E      endp
.text:00B1138E
.text:00B11393
.text:00B11393 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11393
.text:00B11393 ; Attributes: thunk
.text:00B11393
.text:00B11393 sub_B11393      proc near               ; CODE XREF: .text:00B12760↓p
.text:00B11393                 jmp     sub_B13950
.text:00B11393 sub_B11393      endp
.text:00B11393
.text:00B11398 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11398                 jmp     loc_B12500
.text:00B1139D ; ───────────────────────────────────────────────────────────────────────────
.text:00B1139D                 jmp     loc_B15CAF
.text:00B113A2 ; [00000005 BYTES: COLLAPSED FUNCTION j_strcat_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B113A7
.text:00B113A7 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B113A7
.text:00B113A7 ; Attributes: thunk
.text:00B113A7
.text:00B113A7 ; int __cdecl sub_B113A7(int, char *Source)
.text:00B113A7 sub_B113A7      proc near               ; CODE XREF: .text:00B1254D↓p
.text:00B113A7                                         ; sub_B12610+41↓p
.text:00B113A7
.text:00B113A7 Source          = dword ptr  8
.text:00B113A7
.text:00B113A7                 jmp     sub_B12EA0
.text:00B113A7 sub_B113A7      endp
.text:00B113A7
.text:00B113AC ; [00000005 BYTES: COLLAPSED FUNCTION j__CrtDbgReport. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B113B1 ; ───────────────────────────────────────────────────────────────────────────
.text:00B113B1                 jmp     A_func2
.text:00B113B6 ; ───────────────────────────────────────────────────────────────────────────
.text:00B113B6                 jmp     loc_B14B70
.text:00B113BB
.text:00B113BB ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B113BB
.text:00B113BB ; Attributes: thunk
```

```
.text:00B113BB
.text:00B113BB sub_B113BB      proc near               ; CODE XREF: sub_B143B0+21↓p
.text:00B113BB                 jmp     sub_B143A0
.text:00B113BB sub_B113BB      endp
.text:00B113BB
.text:00B113C0 ; [00000005 BYTES: COLLAPSED FUNCTION j_strcpy_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B113C5 ; [00000005 BYTES: COLLAPSED FUNCTION j__exit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B113CA ; [00000005 BYTES: COLLAPSED FUNCTION j__initterm. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B113CF ; [00000005 BYTES: COLLAPSED FUNCTION j___current_exception_context. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B113D4
.text:00B113D4 ; =============== S U B R O U T I N E =======================================
.text:00B113D4
.text:00B113D4 ; Attributes: thunk
.text:00B113D4
.text:00B113D4 sub_B113D4      proc near               ; CODE XREF: .text:00B13645↓p
.text:00B113D4                 jmp     sub_B13650
.text:00B113D4 sub_B113D4      endp
.text:00B113D4
.text:00B113D9 ; ---------------------------------------------------------------------------
.text:00B113D9                 jmp     loc_B15B83
.text:00B113DE ; [00000005 BYTES: COLLAPSED FUNCTION j___vcrt_GetModuleFileNameW. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B113E3 ; ---------------------------------------------------------------------------
.text:00B113E3                 jmp     __CxxFrameHandler3
.text:00B113E8
.text:00B113E8 ; =============== S U B R O U T I N E =======================================
.text:00B113E8
.text:00B113E8 ; Attributes: thunk
.text:00B113E8
.text:00B113E8 sub_B113E8      proc near               ; CODE XREF: sub_B13B00:loc_B13B13↓p
.text:00B113E8                 jmp     sub_B13AF0
.text:00B113E8 sub_B113E8      endp
.text:00B113E8
.text:00B113ED ; [00000005 BYTES: COLLAPSED FUNCTION j__configure_narrow_argv. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B113F2
.text:00B113F2 ; =============== S U B R O U T I N E =======================================
.text:00B113F2
.text:00B113F2 ; Attributes: thunk
.text:00B113F2
.text:00B113F2 sub_B113F2      proc near               ; CODE XREF: sub_B13030+36↓p
.text:00B113F2                 jmp     sub_B138E0
.text:00B113F2 sub_B113F2      endp
.text:00B113F2
.text:00B113F7 ; ---------------------------------------------------------------------------
.text:00B113F7                 jmp     loc_B12120
.text:00B113FC
.text:00B113FC ; =============== S U B R O U T I N E =======================================
.text:00B113FC
.text:00B113FC ; Attributes: thunk
.text:00B113FC
.text:00B113FC sub_B113FC      proc near               ; CODE XREF: sub_B127A0:loc_B12809↓p
.text:00B113FC                 jmp     sub_B14380
.text:00B113FC sub_B113FC      endp
.text:00B113FC
.text:00B11401 ; ---------------------------------------------------------------------------
.text:00B11401                 jmp     loc_B11E00
.text:00B11406 ; ---------------------------------------------------------------------------
.text:00B11406                 jmp     loc_B15CA9
.text:00B1140B
.text:00B1140B ; =============== S U B R O U T I N E =======================================
.text:00B1140B
.text:00B1140B ; Attributes: thunk
.text:00B1140B
.text:00B1140B sub_B1140B      proc near               ; CODE XREF: SEH_413E60+9↓p
.text:00B1140B                 jmp     sub_B15D50
.text:00B1140B sub_B1140B      endp
.text:00B1140B
.text:00B11410 ; [00000005 BYTES: COLLAPSED FUNCTION j__set_fmode. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11415
.text:00B11415 ; =============== S U B R O U T I N E =======================================
.text:00B11415
.text:00B11415 ; Attributes: thunk
.text:00B11415
.text:00B11415 ; int sub_B11415(void)
.text:00B11415 sub_B11415      proc near               ; CODE XREF: sub_B128B0+3↓p
.text:00B11415                 jmp     sub_B14200
```

```
.text:00B11415 sub_B11415      endp
.text:00B11415
.text:00B1141A
.text:00B1141A ; =============== S U B R O U T I N E =======================================
.text:00B1141A
.text:00B1141A ; Attributes: thunk
.text:00B1141A
.text:00B1141A sub_B1141A      proc near               ; CODE XREF: .text:00B13CD5↓p
.text:00B1141A                 jmp     sub_B15D30
.text:00B1141A sub_B1141A      endp
.text:00B1141A
.text:00B1141F ; ---------------------------------------------------------------------------
.text:00B1141F                 jmp     loc_B12496
.text:00B11424 ; [00000005 BYTES: COLLAPSED FUNCTION j__cexit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11429
.text:00B11429 ; =============== S U B R O U T I N E =======================================
.text:00B11429
.text:00B11429 ; Attributes: thunk
.text:00B11429
.text:00B11429 ; int __cdecl sub_B11429(struct _EXCEPTION_POINTERS *ExceptionInfo)
.text:00B11429 sub_B11429      proc near               ; CODE XREF: sub_B134F0+FB↓p
.text:00B11429                                         ; sub_B13650+CA↓p ...
.text:00B11429
.text:00B11429 ExceptionInfo   = dword ptr  4
.text:00B11429
.text:00B11429                 jmp     sub_B134B0
.text:00B11429 sub_B11429      endp
.text:00B11429
.text:00B1142E ; ---------------------------------------------------------------------------
.text:00B1142E                 jmp     loc_B15CC7
.text:00B1142E ; ---------------------------------------------------------------------------
.text:00B11433                 db 0E9h
.text:00B11434                 dd 4D8h, 0BA3E9h, 0E9EE900h, 29E90000h, 0E900000Eh, 474h
.text:00B1144C                 dd 0DDFE9h, 4AFE900h, 5E90000h, 0E9000005h, 9A0h, 0CBBE9h
.text:00B11464                 dd 0A56E900h, 0C1E90000h, 0E900000Bh, 49Dh
.text:00B11474
.text:00B11474 ; =============== S U B R O U T I N E =======================================
.text:00B11474
.text:00B11474 ; Attributes: thunk
.text:00B11474
.text:00B11474 ; int __cdecl printf(char *Format, char)
.text:00B11474 printf          proc near               ; CODE XREF: A_func1+38↓p
.text:00B11474                                         ; A_func2+3F↓p
.text:00B11474
.text:00B11474 Format          = dword ptr  4
.text:00B11474
.text:00B11474                 jmp     sub_B11AC0
.text:00B11474 printf          endp
.text:00B11474
.text:00B11479 ; ---------------------------------------------------------------------------
.text:00B11479                 jmp     loc_B1190B
.text:00B1147E ; ---------------------------------------------------------------------------
.text:00B1147E                 jmp     loc_B11917
.text:00B11483 ; [00000005 BYTES: COLLAPSED FUNCTION j_A_func1. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11488 ; ---------------------------------------------------------------------------
.text:00B11488                 jmp     loc_B1191D
.text:00B1148D ; [00000005 BYTES: COLLAPSED FUNCTION j_A_func2. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11492
.text:00B11492 ; =============== S U B R O U T I N E =======================================
.text:00B11492
.text:00B11492 ; Attributes: thunk
.text:00B11492
.text:00B11492 ; int __cdecl sub_B11492(FILE *Stream, char *Format, _locale_t Locale, va_list ArgList)
.text:00B11492 sub_B11492      proc near               ; CODE XREF: sub_B11AC0+4A↓p
.text:00B11492
.text:00B11492 Stream          = dword ptr  4
.text:00B11492 Format          = dword ptr  8
.text:00B11492 Locale          = dword ptr  0Ch
.text:00B11492 ArgList         = dword ptr  10h
.text:00B11492
.text:00B11492                 jmp     sub_B119A0
.text:00B11492 sub_B11492      endp
.text:00B11492
.text:00B11492 ; ---------------------------------------------------------------------------
.text:00B11497                 db 46Eh dup(0CCh)
```

```
.text:00B11905 ; [00000006 BYTES: COLLAPSED FUNCTION
std::basic_ostream<char,std::char_traits<char>>::operator<<(int). PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1190B ; ───────────────────────────────────────────────────────────────────────────
.text:00B1190B
.text:00B1190B loc_B1190B:                             ; CODE XREF: .text:00B11479↑j
.text:00B1190B                 jmp     ds:__stdio_common_vfprintf
.text:00B11911 ; [00000006 BYTES: COLLAPSED FUNCTION getchar. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B11917 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11917
.text:00B11917 loc_B11917:                             ; CODE XREF: .text:00B1147E↑j
.text:00B11917                 jmp     ds:__acrt_iob_func
.text:00B1191D ; ───────────────────────────────────────────────────────────────────────────
.text:00B1191D
.text:00B1191D loc_B1191D:                             ; CODE XREF: .text:00B11488↑j
.text:00B1191D                 jmp     ds:system
.text:00B1191D ; ───────────────────────────────────────────────────────────────────────────
.text:00B11923                 db 3Dh dup(0CCh)
.text:00B11960 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11960
.text:00B11960 loc_B11960:                             ; CODE XREF: .text:00B11159↑j
.text:00B11960                 push    ebp
.text:00B11961                 mov     ebp, esp
.text:00B11963                 sub     esp, 0C0h
.text:00B11969                 push    ebx
.text:00B1196A                 push    esi
.text:00B1196B                 push    edi
.text:00B1196C                 mov     edi, ebp
.text:00B1196E                 xor     ecx, ecx
.text:00B11970                 mov     eax, 0CCCCCCCCh
.text:00B11975                 rep stosd
.text:00B11977                 mov     ecx, offset one
.text:00B1197C                 call    _GetCurrentThreadId
.text:00B11981                 pop     edi
.text:00B11982                 pop     esi
.text:00B11983                 pop     ebx
.text:00B11984                 add     esp, 0C0h
.text:00B1198A                 cmp     ebp, esp
.text:00B1198C                 call    CGC_1
.text:00B11991                 mov     esp, ebp
.text:00B11993                 pop     ebp
.text:00B11994                 retn
.text:00B11994 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11995                 align 10h
.text:00B119A0
.text:00B119A0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B119A0
.text:00B119A0 ; Attributes: bp-based frame
.text:00B119A0
.text:00B119A0 ; int __cdecl sub_B119A0(FILE *Stream, char *Format, _locale_t Locale, va_list ArgList)
.text:00B119A0 sub_B119A0      proc near               ; CODE XREF: sub_B11492↑j
.text:00B119A0
.text:00B119A0 Stream          = dword ptr  8
.text:00B119A0 Format          = dword ptr  0Ch
.text:00B119A0 Locale          = dword ptr  10h
.text:00B119A0 ArgList         = dword ptr  14h
.text:00B119A0
.text:00B119A0                 push    ebp
.text:00B119A1                 mov     ebp, esp
.text:00B119A3                 sub     esp, 0C0h
.text:00B119A9                 push    ebx
.text:00B119AA                 push    esi
.text:00B119AB                 push    edi
.text:00B119AC                 mov     edi, ebp
.text:00B119AE                 xor     ecx, ecx
.text:00B119B0                 mov     eax, 0CCCCCCCCh
.text:00B119B5                 rep stosd
.text:00B119B7                 mov     ecx, offset unk_B1E01D
.text:00B119BC                 call    _GetCurrentThreadId
.text:00B119C1                 mov     esi, esp
.text:00B119C3                 mov     eax, [ebp+ArgList]
.text:00B119C6                 push    eax             ; ArgList
.text:00B119C7                 mov     ecx, [ebp+Locale]
.text:00B119CA                 push    ecx             ; Locale
.text:00B119CB                 mov     edx, [ebp+Format]
.text:00B119CE                 push    edx             ; Format
```

```
.text:00B119CF                 mov     eax, [ebp+Stream]
.text:00B119D2                 push    eax             ; Stream
.text:00B119D3                 call    sub_B1135C
.text:00B119D8                 mov     ecx, [eax+4]
.text:00B119DB                 push    ecx
.text:00B119DC                 mov     edx, [eax]
.text:00B119DE                 push    edx             ; Options
.text:00B119DF                 call    ds:__stdio_common_vfprintf
.text:00B119E5                 add     esp, 18h
.text:00B119E8                 cmp     esi, esp
.text:00B119EA                 call    CGC_1
.text:00B119EF                 pop     edi
.text:00B119F0                 pop     esi
.text:00B119F1                 pop     ebx
.text:00B119F2                 add     esp, 0C0h
.text:00B119F8                 cmp     ebp, esp
.text:00B119FA                 call    CGC_1
.text:00B119FF                 mov     esp, ebp
.text:00B11A01                 pop     ebp
.text:00B11A02                 retn
.text:00B11A02 sub_B119A0      endp
.text:00B11A02
.text:00B11A02 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11A03                 db 0BDh dup(0CCh)
.text:00B11AC0
.text:00B11AC0 ; =============== S U B R O U T I N E =======================================
.text:00B11AC0
.text:00B11AC0 ; Attributes: bp-based frame
.text:00B11AC0
.text:00B11AC0 ; int __cdecl sub_B11AC0(char *Format, char)
.text:00B11AC0 sub_B11AC0      proc near               ; CODE XREF: printf↑j
.text:00B11AC0
.text:00B11AC0 var_24          = byte ptr -24h
.text:00B11AC0 ArgList         = dword ptr -14h
.text:00B11AC0 var_8           = dword ptr -8
.text:00B11AC0 Format          = dword ptr  8
.text:00B11AC0 arg_4           = byte ptr  0Ch
.text:00B11AC0
.text:00B11AC0                 push    ebp
.text:00B11AC1                 mov     ebp, esp
.text:00B11AC3                 sub     esp, 0E4h
.text:00B11AC9                 push    ebx
.text:00B11ACA                 push    esi
.text:00B11ACB                 push    edi
.text:00B11ACC                 lea     edi, [ebp+var_24]
.text:00B11ACF                 mov     ecx, 9
.text:00B11AD4                 mov     eax, 0CCCCCCCCh
.text:00B11AD9                 rep stosd
.text:00B11ADB                 mov     ecx, offset unk_B1E01D
.text:00B11AE0                 call    _GetCurrentThreadId
.text:00B11AE5                 lea     eax, [ebp+arg_4]
.text:00B11AE8                 mov     [ebp+ArgList], eax
.text:00B11AEB                 mov     eax, [ebp+ArgList]
.text:00B11AEE                 push    eax             ; ArgList
.text:00B11AEF                 push    0               ; Locale
.text:00B11AF1                 mov     ecx, [ebp+Format]
.text:00B11AF4                 push    ecx             ; Format
.text:00B11AF5                 mov     esi, esp
.text:00B11AF7                 push    1               ; Ix
.text:00B11AF9                 call    ds:__acrt_iob_func
.text:00B11AFF                 add     esp, 4
.text:00B11B02                 cmp     esi, esp
.text:00B11B04                 call    CGC_1
.text:00B11B09                 push    eax             ; Stream
.text:00B11B0A                 call    sub_B11492
.text:00B11B0F                 add     esp, 10h
.text:00B11B12                 mov     [ebp+var_8], eax
.text:00B11B15                 mov     [ebp+ArgList], 0
.text:00B11B1C                 mov     eax, [ebp+var_8]
.text:00B11B1F                 pop     edi
.text:00B11B20                 pop     esi
.text:00B11B21                 pop     ebx
.text:00B11B22                 add     esp, 0E4h
.text:00B11B28                 cmp     ebp, esp
.text:00B11B2A                 call    CGC_1
```

```
.text:00B11B2F                 mov     esp, ebp
.text:00B11B31                 pop     ebp
.text:00B11B32                 retn
.text:00B11B32 sub_B11AC0      endp
.text:00B11B32
.text:00B11B32 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11B33                 align 10h
.text:00B11B40
.text:00B11B40 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11B40
.text:00B11B40 ; Attributes: bp-based frame
.text:00B11B40
.text:00B11B40 ; int __cdecl main_0(int argc, const char **argv, const char **envp)
.text:00B11B40 _main_0         proc near               ; CODE XREF: _main↑j
.text:00B11B40
.text:00B11B40 var_14          = byte ptr -14h
.text:00B11B40 var_10          = byte ptr -10h
.text:00B11B40 var_4           = dword ptr -4
.text:00B11B40 argc            = dword ptr  8
.text:00B11B40 argv            = dword ptr  0Ch
.text:00B11B40 envp            = dword ptr  10h
.text:00B11B40
.text:00B11B40                 push    ebp
.text:00B11B41                 mov     ebp, esp
.text:00B11B43                 sub     esp, 0D4h
.text:00B11B49                 push    ebx
.text:00B11B4A                 push    esi
.text:00B11B4B                 push    edi
.text:00B11B4C                 lea     edi, [ebp+var_14]
.text:00B11B4F                 mov     ecx, 5
.text:00B11B54                 mov     eax, 0CCCCCCCCh
.text:00B11B59                 rep stosd
.text:00B11B5B                 mov     eax, ___security_cookie
.text:00B11B60                 xor     eax, ebp
.text:00B11B62                 mov     [ebp+var_4], eax
.text:00B11B65                 mov     ecx, offset one
.text:00B11B6A                 call    _GetCurrentThreadId
.text:00B11B6F                 lea     ecx, [ebp+var_10]
.text:00B11B72                 call    j_A_func1
.text:00B11B77                 lea     ecx, [ebp+var_10]
.text:00B11B7A                 call    j_A_func2
.text:00B11B7F                 mov     esi, esp
.text:00B11B81                 call    ds:__imp_getchar
.text:00B11B87                 cmp     esi, esp
.text:00B11B89                 call    CGC_1
.text:00B11B8E                 xor     eax, eax
.text:00B11B90                 push    edx
.text:00B11B91                 mov     ecx, ebp
.text:00B11B93                 push    eax
.text:00B11B94                 lea     edx, dword_B11BC0
.text:00B11B9A                 call    CGC_2
.text:00B11B9F                 pop     eax
.text:00B11BA0                 pop     edx
.text:00B11BA1                 pop     edi
.text:00B11BA2                 pop     esi
.text:00B11BA3                 pop     ebx
.text:00B11BA4                 mov     ecx, [ebp+var_4]
.text:00B11BA7                 xor     ecx, ebp        ; StackCookie
.text:00B11BA9                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B11BAE                 add     esp, 0D4h
.text:00B11BB4                 cmp     ebp, esp
.text:00B11BB6                 call    CGC_1
.text:00B11BBB                 mov     esp, ebp
.text:00B11BBD                 pop     ebp
.text:00B11BBE                 retn
.text:00B11BBE _main_0         endp
.text:00B11BBE
.text:00B11BBE ; ───────────────────────────────────────────────────────────────────────────
.text:00B11BBF                 align 10h
.text:00B11BC0 dword_B11BC0    dd 1                    ; DATA XREF: _main_0+54↑o
.text:00B11BC4                 dd offset dword_B11BC8
.text:00B11BC8 dword_B11BC8    dd 0FFFFFFF0h, 8        ; DATA XREF: .text:00B11BC4↑o
.text:00B11BD0                 dd offset dword_B11BD4
.text:00B11BD4 dword_B11BD4    dd 0CCCC0061h, 8Ah dup(0CCCCCCCCh)
.text:00B11BD4                                         ; DATA XREF: .text:00B11BD0↑o
```

```
.text:00B11E00 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11E00
.text:00B11E00 loc_B11E00:                             ; CODE XREF: .text:00B111B8↑j
.text:00B11E00                                         ; .text:00B11401↑j
.text:00B11E00                 push    ebp
.text:00B11E01                 mov     ebp, esp
.text:00B11E03                 sub     esp, 0C0h
.text:00B11E09                 push    ebx
.text:00B11E0A                 push    esi
.text:00B11E0B                 push    edi
.text:00B11E0C                 mov     edi, ebp
.text:00B11E0E                 xor     ecx, ecx
.text:00B11E10                 mov     eax, 0CCCCCCCCh
.text:00B11E15                 rep stosd
.text:00B11E17                 mov     ecx, offset one
.text:00B11E1C                 call    _GetCurrentThreadId
.text:00B11E21                 pop     edi
.text:00B11E22                 pop     esi
.text:00B11E23                 pop     ebx
.text:00B11E24                 add     esp, 0C0h
.text:00B11E2A                 cmp     ebp, esp
.text:00B11E2C                 call    CGC_1
.text:00B11E31                 mov     esp, ebp
.text:00B11E33                 pop     ebp
.text:00B11E34                 retn
.text:00B11E34 ; ───────────────────────────────────────────────────────────────────────────
.text:00B11E35                 db 8Bh dup(0CCh)
.text:00B11EC0
.text:00B11EC0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B11EC0
.text:00B11EC0 ; Attributes: bp-based frame
.text:00B11EC0
.text:00B11EC0 ; int __thiscall A_func1(_DWORD *this)
.text:00B11EC0 A_func1         proc near               ; CODE XREF: .text:00B11253↑j
.text:00B11EC0                                         ; j_A_func1↑j
.text:00B11EC0
.text:00B11EC0 var_C           = byte ptr -0Ch
.text:00B11EC0 var_8           = dword ptr -8
.text:00B11EC0
.text:00B11EC0                 push    ebp
.text:00B11EC1                 mov     ebp, esp
.text:00B11EC3                 sub     esp, 0CCh
.text:00B11EC9                 push    ebx
.text:00B11ECA                 push    esi
.text:00B11ECB                 push    edi             ; char
.text:00B11ECC                 push    ecx
.text:00B11ECD                 lea     edi, [ebp+var_C]
.text:00B11ED0                 mov     ecx, 3
.text:00B11ED5                 mov     eax, 0CCCCCCCCh
.text:00B11EDA                 rep stosd
.text:00B11EDC                 pop     ecx
.text:00B11EDD                 mov     [ebp+var_8], ecx
.text:00B11EE0                 mov     ecx, offset one
.text:00B11EE5                 call    _GetCurrentThreadId
.text:00B11EEA                 mov     eax, [ebp+var_8]
.text:00B11EED                 mov     dword ptr [eax], 1Fh
.text:00B11EF3                 push    offset aClassAFunction ; "Class A - Function 1\n"
.text:00B11EF8                 call    printf
.text:00B11EFD                 add     esp, 4
.text:00B11F00                 mov     esi, esp
.text:00B11F02                 push    offset Command  ; "ls"
.text:00B11F07                 call    ds:system
.text:00B11F0D                 add     esp, 4
.text:00B11F10                 cmp     esi, esp
.text:00B11F12                 call    CGC_1
.text:00B11F17                 pop     edi
.text:00B11F18                 pop     esi
.text:00B11F19                 pop     ebx
.text:00B11F1A                 add     esp, 0CCh
.text:00B11F20                 cmp     ebp, esp
.text:00B11F22                 call    CGC_1
.text:00B11F27                 mov     esp, ebp
.text:00B11F29                 pop     ebp
.text:00B11F2A                 retn
.text:00B11F2A A_func1         endp
```

```
.text:00B11F2A
.text:00B11F2A ; ───────────────────────────────────────────────────────────────────────────
.text:00B11F2B                db 105h dup(0CCh)
.text:00B12030
.text:00B12030 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12030
.text:00B12030 ; Attributes: bp-based frame
.text:00B12030
.text:00B12030 A_func2        proc near               ; CODE XREF: .text:00B113B1↑j
.text:00B12030                                        ; j_A_func2↑j
.text:00B12030
.text:00B12030 var_C          = byte ptr -0Ch
.text:00B12030 var_8          = dword ptr -8
.text:00B12030
.text:00B12030                push    ebp
.text:00B12031                mov     ebp, esp
.text:00B12033                sub     esp, 0CCh
.text:00B12039                push    ebx
.text:00B1203A                push    esi
.text:00B1203B                push    edi             ; char
.text:00B1203C                push    ecx
.text:00B1203D                lea     edi, [ebp+var_C]
.text:00B12040                mov     ecx, 3
.text:00B12045                mov     eax, 0CCCCCCCCh
.text:00B1204A                rep stosd
.text:00B1204C                pop     ecx
.text:00B1204D                mov     [ebp+var_8], ecx
.text:00B12050                mov     ecx, offset one
.text:00B12055                call    _GetCurrentThreadId
.text:00B1205A                mov     eax, [ebp+var_8]
.text:00B1205D                movss   xmm0, ds:dword_B19B34
.text:00B12065                movss   dword ptr [eax+4], xmm0
.text:00B1206A                push    offset aClassAFunction_0 ; "Class A - Function 2"
.text:00B1206F                call    printf
.text:00B12074                add     esp, 4
.text:00B12077                mov     esi, esp
.text:00B12079                push    offset aPs      ; "ps"
.text:00B1207E                call    ds:system
.text:00B12084                add     esp, 4
.text:00B12087                cmp     esi, esp
.text:00B12089                call    CGC_1
.text:00B1208E                pop     edi
.text:00B1208F                pop     esi
.text:00B12090                pop     ebx
.text:00B12091                add     esp, 0CCh
.text:00B12097                cmp     ebp, esp
.text:00B12099                call    CGC_1
.text:00B1209E                mov     esp, ebp
.text:00B120A0                pop     ebp
.text:00B120A1                retn
.text:00B120A1 A_func2        endp
.text:00B120A1
.text:00B120A1 ; ───────────────────────────────────────────────────────────────────────────
.text:00B120A2                db 7Eh dup(0CCh)
.text:00B12120 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12120
.text:00B12120 loc_B12120:                            ; CODE XREF: .text:00B11208↑j
.text:00B12120                                        ; .text:00B113F7↑j
.text:00B12120                push    ebp
.text:00B12121                mov     ebp, esp
.text:00B12123                pop     ebp
.text:00B12124                retn
.text:00B12124 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12125                db 347h dup(0CCh)
.text:00B1246C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1246C
.text:00B1246C loc_B1246C:                            ; CODE XREF: .text:00B110DC↑j
.text:00B1246C                jmp     ds:?uncaught_exception@std@@YA_NXZ ; std::uncaught_exception(void)
.text:00B12472 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12472
.text:00B12472 loc_B12472:                            ; CODE XREF: .text:00B110C8↑j
.text:00B12472                jmp     ds:?good@ios_base@std@@QBE_NXZ ; std::ios_base::good(void)
.text:00B12478 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12478
.text:00B12478 loc_B12478:                            ; CODE XREF: .text:00B11069↑j
```

```
.text:00B12478                 jmp     ds:?flags@ios_base@std@@QBEHXZ ; std::ios_base::flags(void)
.text:00B1247E ; ───────────────────────────────────────────────────────────────────────────
.text:00B1247E
.text:00B1247E loc_B1247E:                             ; CODE XREF: .text:00B1108C↑j
.text:00B1247E                 jmp     ds:?width@ios_base@std@@QBE_JXZ ; std::ios_base::width(void)
.text:00B12484 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12484
.text:00B12484 loc_B12484:                             ; CODE XREF: .text:00B112AD↑j
.text:00B12484                 jmp     ds:?width@ios_base@std@@QAE_J_J@Z ; std::ios_base::width(__int64)
.text:00B1248A ; [00000006 BYTES: COLLAPSED FUNCTION
std::basic_streambuf<char,std::char_traits<char>>::sputc(char). PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B12490 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12490
.text:00B12490 loc_B12490:                             ; CODE XREF: .text:00B1114A↑j
.text:00B12490                 jmp     ds:?sputn@?$basic_streambuf@DU?$char_traits@D@std@@@std@@QAE_JPBD_J@Z ;
std::basic_streambuf<char,std::char_traits<char>>::sputn(char const *,__int64)
.text:00B12496 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12496
.text:00B12496 loc_B12496:                             ; CODE XREF: .text:00B1141F↑j
.text:00B12496                 jmp     ds:?setstate@?$basic_ios@DU?$char_traits@D@std@@@std@@QAEXH_N@Z ;
std::basic_ios<char,std::char_traits<char>>::setstate(int,bool)
.text:00B1249C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1249C
.text:00B1249C loc_B1249C:                             ; CODE XREF: .text:00B112A3↑j
.text:00B1249C                 jmp
ds:?tie@?$basic_ios@DU?$char_traits@D@std@@@std@@QBEPAV?$basic_ostream@DU?$char_traits@D@std@@@2@XZ ;
std::basic_ios<char,std::char_traits<char>>::tie(void)
.text:00B124A2 ; ───────────────────────────────────────────────────────────────────────────
.text:00B124A2
.text:00B124A2 loc_B124A2:                             ; CODE XREF: .text:00B112F3↑j
.text:00B124A2                 jmp
ds:?rdbuf@?$basic_ios@DU?$char_traits@D@std@@@std@@QBEPAV?$basic_streambuf@DU?$char_traits@D@std@@@2@XZ ;
std::basic_ios<char,std::char_traits<char>>::rdbuf(void)
.text:00B124A8 ; ───────────────────────────────────────────────────────────────────────────
.text:00B124A8
.text:00B124A8 loc_B124A8:                             ; CODE XREF: .text:00B11203↑j
.text:00B124A8                 jmp     ds:?fill@?$basic_ios@DU?$char_traits@D@std@@@std@@QBEDXZ ;
std::basic_ios<char,std::char_traits<char>>::fill(void)
.text:00B124AE ; ───────────────────────────────────────────────────────────────────────────
.text:00B124AE
.text:00B124AE loc_B124AE:                             ; CODE XREF: .text:00B11325↑j
.text:00B124AE                 jmp     ds:?_Osfx@?$basic_ostream@DU?$char_traits@D@std@@@std@@QAEXXZ ;
std::basic_ostream<char,std::char_traits<char>>::_Osfx(void)
.text:00B124B4 ; ───────────────────────────────────────────────────────────────────────────
.text:00B124B4
.text:00B124B4 loc_B124B4:                             ; CODE XREF: .text:00B11177↑j
.text:00B124B4                 jmp     ds:?flush@?$basic_ostream@DU?$char_traits@D@std@@@std@@QAEAAV12@XZ ;
std::basic_ostream<char,std::char_traits<char>>::flush(void)
.text:00B124B4 ; ───────────────────────────────────────────────────────────────────────────
.text:00B124BA                 align 10h
.text:00B124C0
.text:00B124C0 loc_B124C0:                             ; CODE XREF: .text:00B112E9↑j
.text:00B124C0                 push    ebp
.text:00B124C1                 mov     ebp, esp
.text:00B124C3                 push    ebx
.text:00B124C4                 push    esi
.text:00B124C5                 mov     esi, ecx
.text:00B124C7                 mov     ebx, edx
.text:00B124C9                 test    esi, esi
.text:00B124CB                 jz      short loc_B124EC
.text:00B124CD                 test    ebx, ebx
.text:00B124CF                 jz      short loc_B124EC
.text:00B124D1                 mov     edx, [ebp+8]
.text:00B124D4                 test    edx, edx
.text:00B124D6                 jz      short loc_B124EC
.text:00B124D8                 push    edi
.text:00B124D9                 mov     al, 0CCh
.text:00B124DB                 mov     edi, esi
.text:00B124DD                 mov     ecx, ebx
.text:00B124DF                 rep stosb
.text:00B124E1                 mov     eax, [edx]
.text:00B124E3                 mov     [esi+4], eax
.text:00B124E6                 mov     [esi+0Ch], ebx
.text:00B124E9                 mov     [edx], esi
.text:00B124EB                 pop     edi
```

```
.text:00B124EC
.text:00B124EC loc_B124EC:                             ; CODE XREF: .text:00B124CB↑j
.text:00B124EC                                         ; .text:00B124CF↑j ...
.text:00B124EC                 pop     esi
.text:00B124ED                 pop     ebx
.text:00B124EE                 pop     ebp
.text:00B124EF                 retn    4
.text:00B124EF ; ───────────────────────────────────────────────────────────────────────────
.text:00B124F2                 align 10h
.text:00B12500
.text:00B12500 loc_B12500:                             ; CODE XREF: .text:00B11398↑j
.text:00B12500                 push    ebp
.text:00B12501                 mov     ebp, esp
.text:00B12503                 sub     esp, 8
.text:00B12506                 mov     [ebp-4], ecx
.text:00B12509                 push    esi
.text:00B1250A                 mov     esi, [ebp+8]
.text:00B1250D                 push    edi
.text:00B1250E                 test    edx, edx
.text:00B12510                 jz      short loc_B12561
.text:00B12512                 xor     edi, edi
.text:00B12514                 cmp     [edx], edi
.text:00B12516                 jle     short loc_B12561
.text:00B12518                 push    ebx
.text:00B12519                 xor     ebx, ebx
.text:00B1251B                 mov     esi, edx
.text:00B1251D                 lea     ecx, [ecx+0]
.text:00B12520
.text:00B12520 loc_B12520:                             ; CODE XREF: .text:00B1255B↓j
.text:00B12520                 mov     ecx, [esi+4]
.text:00B12523                 mov     eax, [ebp-4]
.text:00B12526                 mov     edx, [ecx+ebx]
.text:00B12529                 cmp     dword ptr [edx+eax-4], 0CCCCCCCCh
.text:00B12531                 jnz     short loc_B12545
.text:00B12533                 mov     eax, [ecx+ebx+4]
.text:00B12537                 add     eax, edx
.text:00B12539                 mov     edx, [ebp-4]
.text:00B1253C                 cmp     dword ptr [eax+edx], 0CCCCCCCCh
.text:00B12543                 jz      short loc_B12555
.text:00B12545
.text:00B12545 loc_B12545:                             ; CODE XREF: .text:00B12531↑j
.text:00B12545                 push    dword ptr [ecx+ebx+8]
.text:00B12549                 mov     eax, [ebp+4]
.text:00B1254C                 push    eax
.text:00B1254D                 call    sub_B113A7
.text:00B12552                 add     esp, 8
.text:00B12555
.text:00B12555 loc_B12555:                             ; CODE XREF: .text:00B12543↑j
.text:00B12555                 inc     edi
.text:00B12556                 add     ebx, 0Ch
.text:00B12559                 cmp     edi, [esi]
.text:00B1255B                 jl      short loc_B12520
.text:00B1255D                 mov     esi, [ebp+8]
.text:00B12560                 pop     ebx
.text:00B12561
.text:00B12561 loc_B12561:                             ; CODE XREF: .text:00B12510↑j
.text:00B12561                                         ; .text:00B12516↑j
.text:00B12561                 xor     edi, edi
.text:00B12563                 mov     eax, esi
.text:00B12565                 test    esi, esi
.text:00B12567                 jz      short loc_B125CC
.text:00B12569                 lea     esp, [esp+0]
.text:00B12570
.text:00B12570 loc_B12570:                             ; CODE XREF: .text:00B12576↓j
.text:00B12570                 mov     eax, [eax+4]
.text:00B12573                 inc     edi
.text:00B12574                 test    eax, eax
.text:00B12576                 jnz     short loc_B12570
.text:00B12578
.text:00B12578 loc_B12578:                             ; CODE XREF: .text:00B125CA↓j
.text:00B12578                 cmp     dword ptr [esi], 0CCCCCCCCh
.text:00B1257E                 jnz     short loc_B1259B
.text:00B12580                 cmp     dword ptr [esi+14h], 0CCCCCCCCh
.text:00B12587                 jnz     short loc_B1259B
.text:00B12589                 cmp     dword ptr [esi+18h], 0CCCCCCCCh
```

```
.text:00B12590                 jnz     short loc_B1259B
.text:00B12592                 cmp     dword ptr [esi+1Ch], 0CCCCCCCCh
.text:00B12599                 jz      short loc_B125A9
.text:00B1259B
.text:00B1259B loc_B1259B:                             ; CODE XREF: .text:00B1257E↑j
.text:00B1259B                                         ; .text:00B12587↑j ...
.text:00B1259B                 mov     eax, [ebp+4]
.text:00B1259E                 push    edi
.text:00B1259F                 push    esi
.text:00B125A0                 push    eax
.text:00B125A1                 call    sub_B111C7
.text:00B125A6                 add     esp, 0Ch
.text:00B125A9
.text:00B125A9 loc_B125A9:                             ; CODE XREF: .text:00B12599↑j
.text:00B125A9                 mov     eax, [esi+0Ch]
.text:00B125AC                 cmp     dword ptr [eax+esi-4], 0CCCCCCCCh
.text:00B125B4                 jz      short loc_B125C4
.text:00B125B6                 mov     eax, [ebp+4]
.text:00B125B9                 push    edi
.text:00B125BA                 push    esi
.text:00B125BB                 push    eax
.text:00B125BC                 call    sub_B111C7
.text:00B125C1                 add     esp, 0Ch
.text:00B125C4
.text:00B125C4 loc_B125C4:                             ; CODE XREF: .text:00B125B4↑j
.text:00B125C4                 mov     esi, [esi+4]
.text:00B125C7                 dec     edi
.text:00B125C8                 test    esi, esi
.text:00B125CA                 jnz     short loc_B12578
.text:00B125CC
.text:00B125CC loc_B125CC:                             ; CODE XREF: .text:00B12567↑j
.text:00B125CC                 pop     edi
.text:00B125CD                 pop     esi
.text:00B125CE                 mov     esp, ebp
.text:00B125D0                 pop     ebp
.text:00B125D1                 retn    4
.text:00B125D1 ; ---------------------------------------------------------------------------
.text:00B125D4                 db 3Ch dup(0CCh)
.text:00B12610
.text:00B12610 ; =============== S U B R O U T I N E =======================================
.text:00B12610
.text:00B12610 ; Attributes: bp-based frame
.text:00B12610
.text:00B12610 ; int __fastcall sub_B12610(int, int *)
.text:00B12610 sub_B12610      proc near               ; CODE XREF: CGC_2↑j
.text:00B12610
.text:00B12610 var_4           = dword ptr -4
.text:00B12610
.text:00B12610                 push    ebp
.text:00B12611                 mov     ebp, esp
.text:00B12613                 push    ecx
.text:00B12614                 push    ebx
.text:00B12615                 mov     ebx, edx
.text:00B12617                 mov     [ebp+var_4], ecx
.text:00B1261A                 push    esi
.text:00B1261B                 xor     esi, esi
.text:00B1261D                 cmp     [ebx], esi
.text:00B1261F                 jle     short loc_B12662
.text:00B12621                 push    edi
.text:00B12622                 xor     edi, edi
.text:00B12624
.text:00B12624 loc_B12624:                             ; CODE XREF: sub_B12610+4F↓j
.text:00B12624                 mov     ecx, [ebx+4]
.text:00B12627                 mov     eax, [ebp+var_4]
.text:00B1262A                 mov     edx, [ecx+edi]
.text:00B1262D                 cmp     dword ptr [edx+eax-4], 0CCCCCCCCh
.text:00B12635                 jnz     short loc_B12649
.text:00B12637                 mov     eax, [ecx+edi+4]
.text:00B1263B                 add     eax, edx
.text:00B1263D                 mov     edx, [ebp+var_4]
.text:00B12640                 cmp     dword ptr [eax+edx], 0CCCCCCCCh
.text:00B12647                 jz      short loc_B12659
.text:00B12649
.text:00B12649 loc_B12649:                             ; CODE XREF: sub_B12610+25↑j
.text:00B12649                 push    dword ptr [ecx+edi+8] ; Source
```

```
.text:00B1264D                 mov     eax, [ebp+4]
.text:00B12650                 push    eax             ; int
.text:00B12651                 call    sub_B113A7
.text:00B12656                 add     esp, 8
.text:00B12659
.text:00B12659 loc_B12659:                             ; CODE XREF: sub_B12610+37↑j
.text:00B12659                 inc     esi
.text:00B1265A                 add     edi, 0Ch
.text:00B1265D                 cmp     esi, [ebx]
.text:00B1265F                 jl      short loc_B12624
.text:00B12661                 pop     edi
.text:00B12662
.text:00B12662 loc_B12662:                             ; CODE XREF: sub_B12610+F↑j
.text:00B12662                 pop     esi
.text:00B12663                 pop     ebx
.text:00B12664                 mov     esp, ebp
.text:00B12666                 pop     ebp
.text:00B12667                 retn
.text:00B12667 sub_B12610      endp
.text:00B12667
.text:00B12667 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12668                 align 20h
.text:00B12680
.text:00B12680 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12680
.text:00B12680
.text:00B12680 ; void __usercall __spoils<edx,ecx> sub_B12680(char@<zf>)
.text:00B12680 sub_B12680      proc near               ; CODE XREF: CGC_1↑j
.text:00B12680                 jnz     short loc_B12683
.text:00B12682                 retn
.text:00B12683 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12683
.text:00B12683 loc_B12683:                             ; CODE XREF: sub_B12680↑j
.text:00B12683                 push    ebp
.text:00B12684                 mov     ebp, esp
.text:00B12686                 sub     esp, 0
.text:00B12689                 push    eax
.text:00B1268A                 push    edx
.text:00B1268B                 push    ebx
.text:00B1268C                 push    esi
.text:00B1268D                 push    edi
.text:00B1268E                 mov     eax, [ebp+4]
.text:00B12691                 push    0
.text:00B12693                 push    eax
.text:00B12694                 call    sub_B11104
.text:00B12699                 add     esp, 8
.text:00B1269C                 pop     edi
.text:00B1269D                 pop     esi
.text:00B1269E                 pop     ebx
.text:00B1269F                 pop     edx
.text:00B126A0                 pop     eax
.text:00B126A1                 mov     esp, ebp
.text:00B126A3                 pop     ebp
.text:00B126A4                 retn
.text:00B126A4 sub_B12680      endp
.text:00B126A4
.text:00B126A4 ; ───────────────────────────────────────────────────────────────────────────
.text:00B126A5                 align 10h
.text:00B126B0
.text:00B126B0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B126B0
.text:00B126B0 ; Attributes: bp-based frame
.text:00B126B0
.text:00B126B0 ; DWORD __thiscall sub_B126B0(_BYTE *this)
.text:00B126B0 sub_B126B0      proc near               ; CODE XREF: _GetCurrentThreadId↑j
.text:00B126B0
.text:00B126B0 var_8           = dword ptr -8
.text:00B126B0 var_4           = dword ptr -4
.text:00B126B0
.text:00B126B0                 push    ebp
.text:00B126B1                 mov     ebp, esp
.text:00B126B3                 sub     esp, 8
.text:00B126B6                 mov     [ebp+var_4], ecx
.text:00B126B9                 mov     eax, [ebp+var_4]
.text:00B126BC                 mov     [ebp+var_8], eax
```

```
.text:00B126BF                 mov     ecx, [ebp+var_4]
.text:00B126C2                 movzx   edx, byte ptr [ecx]
.text:00B126C5                 test    edx, edx
.text:00B126C7                 jz      short loc_B126E1
.text:00B126C9                 cmp     dword_B1C598, 0
.text:00B126D0                 jz      short loc_B126E1
.text:00B126D2                 call    ds:GetCurrentThreadId
.text:00B126D8                 cmp     dword_B1C598, eax
.text:00B126DE                 jnz     short loc_B126E1
.text:00B126E0                 nop
.text:00B126E1
.text:00B126E1 loc_B126E1:                             ; CODE XREF: sub_B126B0+17↑j
.text:00B126E1                                         ; sub_B126B0+20↑j ...
.text:00B126E1                 mov     esp, ebp
.text:00B126E3                 pop     ebp
.text:00B126E4                 retn
.text:00B126E4 sub_B126B0      endp
.text:00B126E4
.text:00B126E4 ; ───────────────────────────────────────────────────────────────────────────
.text:00B126E5                 align 20h
.text:00B12700 ; [0000000E BYTES: COLLAPSED FUNCTION __security_check_cookie(x). PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B1270E                 align 20h
.text:00B12720
.text:00B12720 loc_B12720:                             ; CODE XREF: .text:00B1119A↑j
.text:00B12720                 mov     eax, offset j__CrtDbgReport
.text:00B12725                 retn
.text:00B12725 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12726                 align 10h
.text:00B12730
.text:00B12730 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12730
.text:00B12730
.text:00B12730 sub_B12730      proc near               ; CODE XREF: sub_B110EB↑j
.text:00B12730                 mov     eax, offset j__CrtDbgReportW
.text:00B12735                 retn
.text:00B12735 sub_B12730      endp
.text:00B12735
.text:00B12735 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12736                 align 10h
.text:00B12740
.text:00B12740 loc_B12740:                             ; CODE XREF: .text:loc_B1128F↑j
.text:00B12740                 cmp     byte_B1C138, 0
.text:00B12747                 jnz     short locret_B12768
.text:00B12749                 push    0
.text:00B1274B                 push    1
.text:00B1274D                 push    0
.text:00B1274F                 push    0
.text:00B12751                 push    0
.text:00B12753                 mov     byte_B1C138, 1
.text:00B1275A                 call    sub_B110EB
.text:00B1275F                 push    eax
.text:00B12760                 call    sub_B11393
.text:00B12765                 add     esp, 18h
.text:00B12768
.text:00B12768 locret_B12768:                          ; CODE XREF: .text:00B12747↑j
.text:00B12768                 retn
.text:00B12768 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12769                 align 20h
.text:00B12780
.text:00B12780 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12780
.text:00B12780
.text:00B12780 sub_B12780      proc near               ; CODE XREF: sub_B11352↑j
.text:00B12780                 push    1
.text:00B12782                 push    1
.text:00B12784                 push    0
.text:00B12786                 push    0
.text:00B12788                 push    0
.text:00B1278A                 call    sub_B110EB
.text:00B1278F                 add     esp, 14h
.text:00B12792                 retn
.text:00B12792 sub_B12780      endp
.text:00B12792
.text:00B12792 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12793                 align 10h
```

```
.text:00B127A0
.text:00B127A0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B127A0
.text:00B127A0 ; Attributes: bp-based frame
.text:00B127A0
.text:00B127A0 sub_B127A0      proc near               ; DATA XREF: .rdata:00B19410↓o
.text:00B127A0                 push    ebp
.text:00B127A1                 mov     ebp, esp
.text:00B127A3                 call    sub_B110CD
.text:00B127A8                 call    sub_B1133E
.text:00B127AD                 call    sub_B1100F
.text:00B127B2                 push    1
.text:00B127B4                 call    sub_B11168
.text:00B127B9                 add     esp, 4
.text:00B127BC                 movzx   eax, al
.text:00B127BF                 test    eax, eax
.text:00B127C1                 jnz     short loc_B127CA
.text:00B127C3                 push    7
.text:00B127C5                 call    sub_B1121C
.text:00B127CA
.text:00B127CA loc_B127CA:                             ; CODE XREF: sub_B127A0+21↑j
.text:00B127CA                 fnclex
.text:00B127CC                 call    sub_B1100A
.text:00B127D1                 push    offset Function ; Function
.text:00B127D6                 call    sub_B1123A
.text:00B127DB                 add     esp, 4
.text:00B127DE                 call    sub_B11258
.text:00B127E3                 test    eax, eax
.text:00B127E5                 jz      short loc_B127EE
.text:00B127E7                 push    7
.text:00B127E9                 call    sub_B1121C
.text:00B127EE
.text:00B127EE loc_B127EE:                             ; CODE XREF: sub_B127A0+45↑j
.text:00B127EE                 call    sub_B110AA
.text:00B127F3                 call    sub_B1125D
.text:00B127F8                 test    eax, eax
.text:00B127FA                 jz      short loc_B12809
.text:00B127FC                 push    offset UserMathErrorFunction ; UserMathErrorFunction
.text:00B12801                 call    j___setusermatherr
.text:00B12806                 add     esp, 4
.text:00B12809
.text:00B12809 loc_B12809:                             ; CODE XREF: sub_B127A0+5A↑j
.text:00B12809                 call    sub_B113FC
.text:00B1280E                 call    sub_B1103C
.text:00B12813                 call    sub_B11389
.text:00B12818                 call    sub_B112D0
.text:00B1281D                 push    eax             ; Flag
.text:00B1281E                 call    j__configthreadlocale
.text:00B12823                 add     esp, 4
.text:00B12826                 call    sub_B1134D
.text:00B1282B                 movzx   ecx, al
.text:00B1282E                 test    ecx, ecx
.text:00B12830                 jz      short loc_B12837
.text:00B12832                 call    sub_B11163
.text:00B12837
.text:00B12837 loc_B12837:                             ; CODE XREF: sub_B127A0+90↑j
.text:00B12837                 call    sub_B112D5
.text:00B1283C                 call    sub_B11267
.text:00B12841                 test    eax, eax
.text:00B12843                 jz      short loc_B1284C
.text:00B12845                 push    7
.text:00B12847                 call    sub_B1121C
.text:00B1284C
.text:00B1284C loc_B1284C:                             ; CODE XREF: sub_B127A0+A3↑j
.text:00B1284C                 xor     eax, eax
.text:00B1284E                 pop     ebp
.text:00B1284F                 retn
.text:00B1284F sub_B127A0      endp ; sp-analysis failed
.text:00B1284F
.text:00B1284F ; ───────────────────────────────────────────────────────────────────────────
.text:00B12850                 align 40h
.text:00B12880
.text:00B12880 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12880
.text:00B12880 ; Attributes: bp-based frame
```

```
.text:00B12880
.text:00B12880 sub_B12880      proc near               ; DATA XREF: .rdata:00B19514↓o
.text:00B12880                 push    ebp
.text:00B12881                 mov     ebp, esp
.text:00B12883                 call    sub_B111A4
.text:00B12888                 xor     eax, eax
.text:00B1288A                 pop     ebp
.text:00B1288B                 retn
.text:00B1288B sub_B12880      endp
.text:00B1288B
.text:00B1288B ; ───────────────────────────────────────────────────────────────────────────
.text:00B1288C                 align 10h
.text:00B12890
.text:00B12890 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12890
.text:00B12890 ; Attributes: bp-based frame
.text:00B12890
.text:00B12890 sub_B12890      proc near               ; DATA XREF: .rdata:00B19104↓o
.text:00B12890                 push    ebp
.text:00B12891                 mov     ebp, esp
.text:00B12893                 call    sub_B1138E
.text:00B12898                 call    sub_B1117C
.text:00B1289D                 push    eax             ; NewMode
.text:00B1289E                 call    j__set_new_mode
.text:00B128A3                 add     esp, 4
.text:00B128A6                 pop     ebp
.text:00B128A7                 retn
.text:00B128A7 sub_B12890      endp
.text:00B128A7
.text:00B128A7 ; ───────────────────────────────────────────────────────────────────────────
.text:00B128A8                 align 10h
.text:00B128B0 ; [0000000F BYTES: COLLAPSED FUNCTION sub_B128B0. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B128BF                 db 11h dup(0CCh)
.text:00B128D0 ; [00000214 BYTES: COLLAPSED FUNCTION sub_B128D0. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B12AE4                 db 8Ch dup(0CCh)
.text:00B12B70
.text:00B12B70 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12B70
.text:00B12B70 ; Attributes: bp-based frame
.text:00B12B70
.text:00B12B70 sub_B12B70      proc near               ; CODE XREF: sub_B11258↑j
.text:00B12B70                 push    ebp
.text:00B12B71                 mov     ebp, esp
.text:00B12B73                 call    sub_B112EE
.text:00B12B78                 push    eax             ; mode
.text:00B12B79                 call    j__configure_narrow_argv
.text:00B12B7E                 add     esp, 4
.text:00B12B81                 pop     ebp
.text:00B12B82                 retn
.text:00B12B82 sub_B12B70      endp
.text:00B12B82
.text:00B12B82 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12B83                 align 10h
.text:00B12B90
.text:00B12B90 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12B90
.text:00B12B90 ; Attributes: bp-based frame
.text:00B12B90
.text:00B12B90 sub_B12B90      proc near               ; CODE XREF: sub_B11163↑j
.text:00B12B90                 push    ebp
.text:00B12B91                 mov     ebp, esp
.text:00B12B93                 call    j__initialize_narrow_environment
.text:00B12B98                 pop     ebp
.text:00B12B99                 retn
.text:00B12B99 sub_B12B90      endp
.text:00B12B99
.text:00B12B99 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12B9A                 align 10h
.text:00B12BA0 ; [0000003A BYTES: COLLAPSED FUNCTION sub_B12BA0. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B12BDA                 db 16h dup(0CCh)
.text:00B12BF0
.text:00B12BF0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12BF0
.text:00B12BF0 ; Attributes: bp-based frame
.text:00B12BF0
```

```
.text:00B12BF0 sub_B12BF0      proc near               ; CODE XREF: sub_B110CD↑j
.text:00B12BF0                 push    ebp
.text:00B12BF1                 mov     ebp, esp
.text:00B12BF3                 push    1               ; Type
.text:00B12BF5                 call    j__set_app_type
.text:00B12BFA                 add     esp, 4
.text:00B12BFD                 pop     ebp
.text:00B12BFE                 retn
.text:00B12BFE sub_B12BF0      endp
.text:00B12BFE
.text:00B12BFE ; ───────────────────────────────────────────────────────────────────────────
.text:00B12BFF                 db 11h dup(0CCh)
.text:00B12C10
.text:00B12C10 ; =============== S U B R O U T I N E =======================================
.text:00B12C10
.text:00B12C10 ; Attributes: bp-based frame
.text:00B12C10
.text:00B12C10 sub_B12C10      proc near               ; CODE XREF: sub_B1100F↑j
.text:00B12C10
.text:00B12C10 var_4           = dword ptr -4
.text:00B12C10
.text:00B12C10                 push    ebp
.text:00B12C11                 mov     ebp, esp
.text:00B12C13                 push    ecx
.text:00B12C14                 call    sub_B1114F
.text:00B12C19                 mov     [ebp+var_4], eax
.text:00B12C1C                 call    j___p__commode
.text:00B12C21                 mov     ecx, [ebp+var_4]
.text:00B12C24                 mov     [eax], ecx
.text:00B12C26                 mov     esp, ebp
.text:00B12C28                 pop     ebp
.text:00B12C29                 retn
.text:00B12C29 sub_B12C10      endp
.text:00B12C29
.text:00B12C29 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12C2A                 align 10h
.text:00B12C30
.text:00B12C30 ; =============== S U B R O U T I N E =======================================
.text:00B12C30
.text:00B12C30 ; Attributes: bp-based frame
.text:00B12C30
.text:00B12C30 sub_B12C30      proc near               ; CODE XREF: sub_B1133E↑j
.text:00B12C30                 push    ebp
.text:00B12C31                 mov     ebp, esp
.text:00B12C33                 call    sub_B110FA
.text:00B12C38                 push    eax             ; Mode
.text:00B12C39                 call    j__set_fmode
.text:00B12C3E                 add     esp, 4
.text:00B12C41                 pop     ebp
.text:00B12C42                 retn
.text:00B12C42 sub_B12C30      endp
.text:00B12C42
.text:00B12C42 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12C43                 align 10h
.text:00B12C50 ; [0000000A BYTES: COLLAPSED FUNCTION start_0. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B12C5A                 align 10h
.text:00B12C60
.text:00B12C60 ; =============== S U B R O U T I N E =======================================
.text:00B12C60
.text:00B12C60 ; Attributes: bp-based frame
.text:00B12C60
.text:00B12C60 sub_B12C60      proc near               ; CODE XREF: sub_B13030+9A↓p
.text:00B12C60
.text:00B12C60 Arguments       = dword ptr -1Ch
.text:00B12C60 var_18          = dword ptr -18h
.text:00B12C60 var_14          = dword ptr -14h
.text:00B12C60 var_1           = byte ptr -1
.text:00B12C60 arg_0           = dword ptr  8
.text:00B12C60
.text:00B12C60                 push    ebp
.text:00B12C61                 mov     ebp, esp
.text:00B12C63                 sub     esp, 1Ch
.text:00B12C66                 mov     eax, [ebp+arg_0]
.text:00B12C69                 mov     [ebp+var_18], eax
.text:00B12C6C                 lea     eax, [ebp+var_1]
```

```
.text:00B12C6F                 mov     [ebp+var_14], eax
.text:00B12C72                 lea     eax, [ebp+Arguments]
.text:00B12C75                 push    eax             ; lpArguments
.text:00B12C76                 mov     [ebp+var_1], 0
.text:00B12C7A                 mov     [ebp+Arguments], 1001h
.text:00B12C81                 call    sub_B132B0
.text:00B12C86                 add     esp, 4
.text:00B12C89                 cmp     [ebp+var_1], 0
.text:00B12C8D                 setnz   al
.text:00B12C90                 mov     esp, ebp
.text:00B12C92                 pop     ebp
.text:00B12C93                 retn
.text:00B12C93 sub_B12C60      endp
.text:00B12C93
.text:00B12C93 ; ---------------------------------------------------------------------------
.text:00B12C94                 db 1Ch dup(0CCh)
.text:00B12CB0
.text:00B12CB0 ; =============== S U B R O U T I N E =======================================
.text:00B12CB0
.text:00B12CB0 ; Attributes: bp-based frame
.text:00B12CB0
.text:00B12CB0 sub_B12CB0      proc near               ; CODE XREF: sub_B13030+B6↓p
.text:00B12CB0
.text:00B12CB0 Arguments       = dword ptr -1Ch
.text:00B12CB0 var_18          = dword ptr -18h
.text:00B12CB0 var_14          = dword ptr -14h
.text:00B12CB0 var_10          = dword ptr -10h
.text:00B12CB0 var_C           = dword ptr -0Ch
.text:00B12CB0 var_8           = dword ptr -8
.text:00B12CB0 var_1           = byte ptr -1
.text:00B12CB0 arg_0           = dword ptr  8
.text:00B12CB0 arg_4           = dword ptr  0Ch
.text:00B12CB0 arg_8           = dword ptr  10h
.text:00B12CB0 arg_C           = dword ptr  14h
.text:00B12CB0
.text:00B12CB0                 push    ebp
.text:00B12CB1                 mov     ebp, esp
.text:00B12CB3                 sub     esp, 1Ch
.text:00B12CB6                 mov     eax, [ebp+arg_0]
.text:00B12CB9                 mov     [ebp+var_18], eax
.text:00B12CBC                 mov     eax, [ebp+arg_4]
.text:00B12CBF                 mov     [ebp+var_14], eax
.text:00B12CC2                 mov     eax, [ebp+arg_8]
.text:00B12CC5                 mov     [ebp+var_10], eax
.text:00B12CC8                 lea     eax, [ebp+var_1]
.text:00B12CCB                 mov     [ebp+var_C], eax
.text:00B12CCE                 mov     eax, [ebp+arg_C]
.text:00B12CD1                 mov     [ebp+var_8], eax
.text:00B12CD4                 lea     eax, [ebp+Arguments]
.text:00B12CD7                 push    eax             ; lpArguments
.text:00B12CD8                 mov     [ebp+var_1], 0
.text:00B12CDC                 mov     [ebp+Arguments], 1002h
.text:00B12CE3                 call    sub_B132B0
.text:00B12CE8                 add     esp, 4
.text:00B12CEB                 cmp     [ebp+var_1], 0
.text:00B12CEF                 setnz   al
.text:00B12CF2                 mov     esp, ebp
.text:00B12CF4                 pop     ebp
.text:00B12CF5                 retn
.text:00B12CF5 sub_B12CB0      endp
.text:00B12CF5
.text:00B12CF5 ; ---------------------------------------------------------------------------
.text:00B12CF6                 db 1Ah dup(0CCh)
.text:00B12D10
.text:00B12D10 ; =============== S U B R O U T I N E =======================================
.text:00B12D10
.text:00B12D10 ; Attributes: bp-based frame
.text:00B12D10
.text:00B12D10 sub_B12D10      proc near               ; CODE XREF: sub_B111C7↑j
.text:00B12D10
.text:00B12D10 var_144         = dword ptr -144h
.text:00B12D10 Buffer          = byte ptr -140h
.text:00B12D10 var_4C          = byte ptr -4Ch
.text:00B12D10 var_18          = byte ptr -18h
.text:00B12D10 var_4           = dword ptr -4
```

```
.text:00B12D10 arg_0           = dword ptr  8
.text:00B12D10 arg_4           = dword ptr  0Ch
.text:00B12D10 arg_8           = dword ptr  10h
.text:00B12D10
.text:00B12D10                 push    ebp
.text:00B12D11                 mov     ebp, esp
.text:00B12D13                 sub     esp, 144h
.text:00B12D19                 mov     eax, ___security_cookie
.text:00B12D1E                 xor     eax, ebp
.text:00B12D20                 mov     [ebp+var_4], eax
.text:00B12D23                 mov     eax, [ebp+arg_0]
.text:00B12D26                 push    ebx
.text:00B12D27                 mov     ebx, [ebp+arg_4]
.text:00B12D2A                 push    edi
.text:00B12D2B                 mov     edi, dword_B1C01C
.text:00B12D31                 mov     [ebp+var_144], eax
.text:00B12D37                 cmp     edi, 0FFFFFFFFh
.text:00B12D3A                 jz      loc_B12DE6
.text:00B12D40                 test    ebx, ebx
.text:00B12D42                 jnz     short loc_B12D65
.text:00B12D44                 push    offset MultiByteStr ; "Stack area around _alloca memory reserv"...
.text:00B12D49                 push    4               ; int
.text:00B12D4B                 push    edi             ; int
.text:00B12D4C                 push    eax             ; int
.text:00B12D4D                 call    sub_B13030
.text:00B12D52                 add     esp, 10h
.text:00B12D55                 pop     edi
.text:00B12D56                 pop     ebx
.text:00B12D57                 mov     ecx, [ebp+var_4]
.text:00B12D5A                 xor     ecx, ebp        ; StackCookie
.text:00B12D5C                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B12D61                 mov     esp, ebp
.text:00B12D63                 pop     ebp
.text:00B12D64                 retn
.text:00B12D65 ; ---------------------------------------------------------------------------
.text:00B12D65
.text:00B12D65 loc_B12D65:                             ; CODE XREF: sub_B12D10+32↑j
.text:00B12D65                 mov     eax, [ebx+0Ch]
.text:00B12D68                 push    esi
.text:00B12D69                 sub     eax, 24h ; '$'
.text:00B12D6C                 lea     esi, [ebx+20h]
.text:00B12D6F                 push    eax
.text:00B12D70                 push    esi
.text:00B12D71                 lea     eax, [ebp+var_4C]
.text:00B12D74                 push    eax
.text:00B12D75                 lea     eax, [ebp+var_18]
.text:00B12D78                 push    eax
.text:00B12D79                 call    sub_B12F80
.text:00B12D7E                 push    offset asc_B1A0E0 ; "\n"
.text:00B12D83                 lea     eax, [ebp+var_4C]
.text:00B12D86                 push    eax
.text:00B12D87                 push    offset asc_B1A0E4 ; "> "
.text:00B12D8C                 lea     eax, [ebp+var_18]
.text:00B12D8F                 push    eax
.text:00B12D90                 mov     eax, [ebx+0Ch]
.text:00B12D93                 push    offset aData    ; "\nData: <"
.text:00B12D98                 push    [ebp+arg_8]
.text:00B12D9B                 sub     eax, 24h ; '$'
.text:00B12D9E                 push    offset aAllocationNumb ; "\nAllocation number within this functio"...
.text:00B12DA3                 push    eax
.text:00B12DA4                 push    offset aSize    ; "\nSize: "
.text:00B12DA9                 push    esi
.text:00B12DAA                 push    offset aAddress0x ; "\nAddress: 0x"
.text:00B12DAF                 push    offset aStackAreaAroun_0 ; "Stack area around _alloca memory reserv"...
.text:00B12DB4                 push    offset Format   ; "%s%s%p%s%zd%s%d%s%s%s%s%s"
.text:00B12DB9                 lea     eax, [ebp+Buffer]
.text:00B12DBF                 push    0F4h            ; BufferCount
.text:00B12DC4                 push    eax             ; Buffer
.text:00B12DC5                 call    sub_B11334
.text:00B12DCA                 add     esp, 4Ch
.text:00B12DCD                 lea     eax, [ebp+Buffer]
.text:00B12DD3                 push    eax             ; lpMultiByteStr
.text:00B12DD4                 push    4               ; int
.text:00B12DD6                 push    edi             ; int
.text:00B12DD7                 push    [ebp+var_144]   ; int
```

```
.text:00B12DDD                 call    sub_B13030
.text:00B12DE2                 add     esp, 10h
.text:00B12DE5                 pop     esi
.text:00B12DE6
.text:00B12DE6 loc_B12DE6:                             ; CODE XREF: sub_B12D10+2A↑j
.text:00B12DE6                 mov     ecx, [ebp+var_4]
.text:00B12DE9                 pop     edi
.text:00B12DEA                 xor     ecx, ebp        ; StackCookie
.text:00B12DEC                 pop     ebx
.text:00B12DED                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B12DF2                 mov     esp, ebp
.text:00B12DF4                 pop     ebp
.text:00B12DF5                 retn
.text:00B12DF5 sub_B12D10      endp
.text:00B12DF5
.text:00B12DF5 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12DF6                 db 3Ah dup(0CCh)
.text:00B12E30
.text:00B12E30 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12E30
.text:00B12E30 ; Attributes: bp-based frame
.text:00B12E30
.text:00B12E30 ; int __cdecl sub_B12E30(int, unsigned int)
.text:00B12E30 sub_B12E30      proc near               ; CODE XREF: sub_B11104↑j
.text:00B12E30
.text:00B12E30 arg_0           = dword ptr  8
.text:00B12E30 arg_4           = dword ptr  0Ch
.text:00B12E30
.text:00B12E30                 push    ebp
.text:00B12E31                 mov     ebp, esp
.text:00B12E33                 mov     eax, [ebp+arg_4]
.text:00B12E36                 cmp     eax, 4
.text:00B12E39                 ja      short loc_B12E5E
.text:00B12E3B                 mov     ecx, dword_B1C00C[eax*4]
.text:00B12E42                 mov     edx, ds:lpMultiByteStr[eax*4]
.text:00B12E49                 cmp     ecx, 0FFFFFFFFh
.text:00B12E4C                 jz      short loc_B12E7C
.text:00B12E4E                 push    edx             ; lpMultiByteStr
.text:00B12E4F                 push    eax             ; int
.text:00B12E50                 push    ecx             ; int
.text:00B12E51                 push    [ebp+arg_0]     ; int
.text:00B12E54                 call    sub_B13030
.text:00B12E59                 add     esp, 10h
.text:00B12E5C                 pop     ebp
.text:00B12E5D                 retn
.text:00B12E5E ; ───────────────────────────────────────────────────────────────────────────
.text:00B12E5E
.text:00B12E5E loc_B12E5E:                             ; CODE XREF: sub_B12E30+9↑j
.text:00B12E5E                 mov     edx, ds:off_B19B54 ; "Unknown Runtime Check Error\r\n"
.text:00B12E64                 mov     eax, 5
.text:00B12E69                 push    edx             ; lpMultiByteStr
.text:00B12E6A                 push    eax             ; int
.text:00B12E6B                 mov     ecx, 1
.text:00B12E70                 push    ecx             ; int
.text:00B12E71                 push    [ebp+arg_0]     ; int
.text:00B12E74                 call    sub_B13030
.text:00B12E79                 add     esp, 10h
.text:00B12E7C
.text:00B12E7C loc_B12E7C:                             ; CODE XREF: sub_B12E30+1C↑j
.text:00B12E7C                 pop     ebp
.text:00B12E7D                 retn
.text:00B12E7D sub_B12E30      endp
.text:00B12E7D
.text:00B12E7D ; ───────────────────────────────────────────────────────────────────────────
.text:00B12E7E                 db 22h dup(0CCh)
.text:00B12EA0
.text:00B12EA0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12EA0
.text:00B12EA0 ; Attributes: bp-based frame
.text:00B12EA0
.text:00B12EA0 ; void __cdecl sub_B12EA0(int, char *Source)
.text:00B12EA0 sub_B12EA0      proc near               ; CODE XREF: sub_B113A7↑j
.text:00B12EA0
.text:00B12EA0 Destination     = byte ptr -404h
.text:00B12EA0 var_4           = dword ptr -4
```

```
.text:00B12EA0 arg_0           = dword ptr  8
.text:00B12EA0 Source          = dword ptr  0Ch
.text:00B12EA0
.text:00B12EA0                 push    ebp
.text:00B12EA1                 mov     ebp, esp
.text:00B12EA3                 sub     esp, 404h
.text:00B12EA9                 mov     eax, ___security_cookie
.text:00B12EAE                 xor     eax, ebp
.text:00B12EB0                 mov     [ebp+var_4], eax
.text:00B12EB3                 push    ebx
.text:00B12EB4                 mov     ebx, [ebp+arg_0]
.text:00B12EB7                 push    esi
.text:00B12EB8                 mov     esi, [ebp+Source]
.text:00B12EBB                 push    edi
.text:00B12EBC                 mov     edi, dword_B1C014
.text:00B12EC2                 cmp     edi, 0FFFFFFFFh
.text:00B12EC5                 jz      short loc_B12F3A
.text:00B12EC7                 cmp     byte ptr [esi], 0
.text:00B12ECA                 jz      short loc_B12F28
.text:00B12ECC                 push    esi
.text:00B12ECD                 call    sub_B13010
.text:00B12ED2                 add     eax, 2Dh ; '-'
.text:00B12ED5                 add     esp, 4
.text:00B12ED8                 cmp     eax, 400h
.text:00B12EDD                 ja      short loc_B12F28
.text:00B12EDF                 push    offset Source   ; "Stack around the variable '"
.text:00B12EE4                 lea     eax, [ebp+Destination]
.text:00B12EEA                 push    400h            ; SizeInBytes
.text:00B12EEF                 push    eax             ; Destination
.text:00B12EF0                 call    j_strcpy_s
.text:00B12EF5                 push    esi             ; Source
.text:00B12EF6                 lea     eax, [ebp+Destination]
.text:00B12EFC                 push    400h            ; SizeInBytes
.text:00B12F01                 push    eax             ; Destination
.text:00B12F02                 call    j_strcat_s
.text:00B12F07                 push    offset aWasCorrupted ; "' was corrupted."
.text:00B12F0C                 lea     eax, [ebp+Destination]
.text:00B12F12                 push    400h            ; SizeInBytes
.text:00B12F17                 push    eax             ; Destination
.text:00B12F18                 call    j_strcat_s
.text:00B12F1D                 add     esp, 24h
.text:00B12F20                 lea     eax, [ebp+Destination]
.text:00B12F26                 jmp     short loc_B12F2D
.text:00B12F28 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12F28
.text:00B12F28 loc_B12F28:                             ; CODE XREF: sub_B12EA0+2A↑j
.text:00B12F28                                         ; sub_B12EA0+3D↑j
.text:00B12F28                 mov     eax, offset aStackCorrupted ; "Stack corrupted near unknown variable"
.text:00B12F2D
.text:00B12F2D loc_B12F2D:                             ; CODE XREF: sub_B12EA0+86↑j
.text:00B12F2D                 push    eax             ; lpMultiByteStr
.text:00B12F2E                 push    2               ; int
.text:00B12F30                 push    edi             ; int
.text:00B12F31                 push    ebx             ; int
.text:00B12F32                 call    sub_B13030
.text:00B12F37                 add     esp, 10h
.text:00B12F3A
.text:00B12F3A loc_B12F3A:                             ; CODE XREF: sub_B12EA0+25↑j
.text:00B12F3A                 mov     ecx, [ebp+var_4]
.text:00B12F3D                 pop     edi
.text:00B12F3E                 pop     esi
.text:00B12F3F                 xor     ecx, ebp        ; StackCookie
.text:00B12F41                 pop     ebx
.text:00B12F42                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B12F47                 mov     esp, ebp
.text:00B12F49                 pop     ebp
.text:00B12F4A                 retn
.text:00B12F4A sub_B12EA0      endp
.text:00B12F4A
.text:00B12F4A ; ───────────────────────────────────────────────────────────────────────────
.text:00B12F4B                 align 40h
.text:00B12F80
.text:00B12F80 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B12F80
.text:00B12F80 ; Attributes: bp-based frame
```

```
.text:00B12F80
.text:00B12F80 sub_B12F80      proc near               ; CODE XREF: sub_B12D10+69↑p
.text:00B12F80
.text:00B12F80 var_4           = dword ptr -4
.text:00B12F80 arg_0           = dword ptr  8
.text:00B12F80 arg_4           = dword ptr  0Ch
.text:00B12F80 arg_8           = dword ptr  10h
.text:00B12F80 arg_C           = dword ptr  14h
.text:00B12F80
.text:00B12F80                 push    ebp
.text:00B12F81                 mov     ebp, esp
.text:00B12F83                 push    ecx
.text:00B12F84                 push    ebx
.text:00B12F85                 mov     ebx, [ebp+arg_8]
.text:00B12F88                 xor     edx, edx
.text:00B12F8A                 push    esi
.text:00B12F8B                 mov     esi, [ebp+arg_0]
.text:00B12F8E                 push    edi
.text:00B12F8F                 xor     edi, edi
.text:00B12F91                 sub     ebx, esi
.text:00B12F93                 mov     [ebp+arg_8], ebx
.text:00B12F96
.text:00B12F96 loc_B12F96:                             ; CODE XREF: sub_B12F80+59↓j
.text:00B12F96                 mov     eax, [ebp+arg_C]
.text:00B12F99                 mov     [ebp+var_4], edx
.text:00B12F9C                 cmp     eax, 10h
.text:00B12F9F                 jb      short loc_B12FA6
.text:00B12FA1                 mov     eax, 10h
.text:00B12FA6
.text:00B12FA6 loc_B12FA6:                             ; CODE XREF: sub_B12F80+1F↑j
.text:00B12FA6                 mov     ecx, [ebp+arg_4]
.text:00B12FA9                 add     ecx, edi
.text:00B12FAB                 cmp     edx, eax
.text:00B12FAD                 jnb     short loc_B12FDB
.text:00B12FAF                 mov     bl, [ebx+esi]
.text:00B12FB2                 movzx   eax, bl
.text:00B12FB5                 push    eax             ; ArgList
.text:00B12FB6                 mov     eax, 31h ; '1'
.text:00B12FBB                 push    offset a2x      ; "%.2X "
.text:00B12FC0                 sub     eax, edi
.text:00B12FC2                 push    eax             ; BufferCount
.text:00B12FC3                 push    ecx             ; Buffer
.text:00B12FC4                 call    sub_B11334
.text:00B12FC9                 mov     edx, [ebp+var_4]
.text:00B12FCC                 add     esp, 10h
.text:00B12FCF                 mov     [esi], bl
.text:00B12FD1                 inc     edx
.text:00B12FD2                 mov     ebx, [ebp+arg_8]
.text:00B12FD5                 inc     esi
.text:00B12FD6                 add     edi, 3
.text:00B12FD9                 jmp     short loc_B12F96
.text:00B12FDB ; ───────────────────────────────────────────────────────────────────────────
.text:00B12FDB
.text:00B12FDB loc_B12FDB:                             ; CODE XREF: sub_B12F80+2D↑j
.text:00B12FDB                 pop     edi
.text:00B12FDC                 mov     byte ptr [esi], 0
.text:00B12FDF                 pop     esi
.text:00B12FE0                 mov     byte ptr [ecx], 0
.text:00B12FE3                 pop     ebx
.text:00B12FE4                 mov     esp, ebp
.text:00B12FE6                 pop     ebp
.text:00B12FE7                 retn
.text:00B12FE7 sub_B12F80      endp
.text:00B12FE7
.text:00B12FE7 ; ───────────────────────────────────────────────────────────────────────────
.text:00B12FE8                 db 28h dup(0CCh)
.text:00B13010
.text:00B13010 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13010
.text:00B13010 ; Attributes: bp-based frame
.text:00B13010
.text:00B13010 sub_B13010      proc near               ; CODE XREF: sub_B12EA0+2D↑p
.text:00B13010                                         ; .text:00B13378↓p
.text:00B13010
.text:00B13010 arg_0           = dword ptr  8
```

```
.text:00B13010
.text:00B13010                 push    ebp
.text:00B13011                 mov     ebp, esp
.text:00B13013                 mov     edx, [ebp+arg_0]
.text:00B13016                 mov     eax, edx
.text:00B13018
.text:00B13018 loc_B13018:                             ; CODE XREF: sub_B13010+D↓j
.text:00B13018                 mov     cl, [eax]
.text:00B1301A                 inc     eax
.text:00B1301B                 test    cl, cl
.text:00B1301D                 jnz     short loc_B13018
.text:00B1301F                 sub     eax, edx
.text:00B13021                 dec     eax
.text:00B13022                 pop     ebp
.text:00B13023                 retn
.text:00B13023 sub_B13010      endp
.text:00B13023
.text:00B13023 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13024                 align 10h
.text:00B13030
.text:00B13030 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13030
.text:00B13030 ; Attributes: bp-based frame
.text:00B13030
.text:00B13030 ; int __cdecl sub_B13030(int, int, int, LPCCH lpMultiByteStr)
.text:00B13030 sub_B13030      proc near               ; CODE XREF: sub_B12D10+3D↑p
.text:00B13030                                         ; sub_B12D10+CD↑p  ...
.text:00B13030
.text:00B13030 var_E3C         = dword ptr -0E3Ch
.text:00B13030 var_E38         = dword ptr -0E38h
.text:00B13030 var_E34         = dword ptr -0E34h
.text:00B13030 var_E30         = dword ptr -0E30h
.text:00B13030 WideCharStr     = word ptr -0E2Ch
.text:00B13030 var_A2C         = byte ptr -0A2Ch
.text:00B13030 MultiByteStr    = byte ptr -720h
.text:00B13030 var_414         = word ptr -414h
.text:00B13030 var_20C         = word ptr -20Ch
.text:00B13030 var_4           = dword ptr -4
.text:00B13030 arg_0           = dword ptr  8
.text:00B13030 arg_4           = dword ptr  0Ch
.text:00B13030 arg_8           = dword ptr  10h
.text:00B13030 lpMultiByteStr  = dword ptr  14h
.text:00B13030
.text:00B13030                 push    ebp
.text:00B13031                 mov     ebp, esp
.text:00B13033                 sub     esp, 0E3Ch
.text:00B13039                 mov     eax, ___security_cookie
.text:00B1303E                 xor     eax, ebp
.text:00B13040                 mov     [ebp+var_4], eax
.text:00B13043                 mov     eax, [ebp+arg_8]
.text:00B13046                 push    ebx
.text:00B13047                 mov     ebx, [ebp+lpMultiByteStr]
.text:00B1304A                 push    esi
.text:00B1304B                 mov     esi, [ebp+arg_0]
.text:00B1304E                 push    edi
.text:00B1304F                 push    esi
.text:00B13050                 mov     [ebp+var_E30], eax
.text:00B13056                 mov     [ebp+var_E3C], ebx
.text:00B1305C                 mov     [ebp+var_E34], 0
.text:00B13066                 call    sub_B113F2
.text:00B1306B                 mov     edi, eax
.text:00B1306D                 add     esp, 4
.text:00B13070                 test    edi, edi
.text:00B13072                 jnz     short loc_B13083
.text:00B13074                 push    esi
.text:00B13075                 call    sub_B110A0
.text:00B1307A                 add     esp, 4
.text:00B1307D                 mov     [ebp+var_E34], eax
.text:00B13083
.text:00B13083 loc_B13083:                             ; CODE XREF: sub_B13030+42↑j
.text:00B13083                 push    0               ; cchWideChar
.text:00B13085                 push    0               ; lpWideCharStr
.text:00B13087                 push    0FFFFFFFFh      ; cbMultiByte
.text:00B13089                 push    ebx             ; lpMultiByteStr
.text:00B1308A                 push    0               ; dwFlags
```

```
.text:00B1308C                 push    0FDE9h          ; CodePage
.text:00B13091                 call    ds:MultiByteToWideChar
.text:00B13097                 cmp     eax, 200h
.text:00B1309C                 jnb     short loc_B130C0
.text:00B1309E                 push    eax             ; cchWideChar
.text:00B1309F                 lea     eax, [ebp+WideCharStr]
.text:00B130A5                 push    eax             ; lpWideCharStr
.text:00B130A6                 push    0FFFFFFFFh      ; cbMultiByte
.text:00B130A8                 push    ebx             ; lpMultiByteStr
.text:00B130A9                 push    0               ; dwFlags
.text:00B130AB                 push    0FDE9h          ; CodePage
.text:00B130B0                 call    ds:MultiByteToWideChar
.text:00B130B6                 lea     ebx, [ebp+WideCharStr]
.text:00B130BC                 test    eax, eax
.text:00B130BE                 jnz     short loc_B130C5
.text:00B130C0
.text:00B130C0 loc_B130C0:                             ; CODE XREF: sub_B13030+6C↑j
.text:00B130C0                 mov     ebx, offset aRuntimeCheckEr ; "Runtime Check Error.\r\n Unable to disp" ...
.text:00B130C5
.text:00B130C5 loc_B130C5:                             ; CODE XREF: sub_B13030+8E↑j
.text:00B130C5                 push    1002h
.text:00B130CA                 call    sub_B12C60
.text:00B130CF                 add     esp, 4
.text:00B130D2                 test    al, al
.text:00B130D4                 jz      short loc_B130F8
.text:00B130D6                 mov     eax, [ebp+var_E30]
.text:00B130DC                 push    ebx
.text:00B130DD                 push    esi
.text:00B130DE                 push    ds:dword_B19B58[eax*4]
.text:00B130E5                 push    eax
.text:00B130E6                 call    sub_B12CB0
.text:00B130EB                 add     esp, 10h
.text:00B130EE                 test    al, al
.text:00B130F0                 jnz     loc_B1321D
.text:00B130F6                 jmp     short loc_B130FA
.text:00B130F8 ; ---------------------------------------------------------------------------
.text:00B130F8
.text:00B130F8 loc_B130F8:                             ; CODE XREF: sub_B13030+A4↑j
.text:00B130F8                 mov     al, 1
.text:00B130FA
.text:00B130FA loc_B130FA:                             ; CODE XREF: sub_B13030+C6↑j
.text:00B130FA                 cmp     [ebp+var_E34], 0
.text:00B13101                 jnz     short loc_B1310B
.text:00B13103                 test    edi, edi
.text:00B13105                 jz      loc_B1321C
.text:00B1310B
.text:00B1310B loc_B1310B:                             ; CODE XREF: sub_B13030+D1↑j
.text:00B1310B                 test    al, al
.text:00B1310D                 jz      short loc_B1311D
.text:00B1310F                 call    ds:IsDebuggerPresent
.text:00B13115                 test    eax, eax
.text:00B13117                 jnz     loc_B1321C
.text:00B1311D
.text:00B1311D loc_B1311D:                             ; CODE XREF: sub_B13030+DD↑j
.text:00B1311D                 push    104h
.text:00B13122                 lea     eax, [ebp+var_414]
.text:00B13128                 push    eax
.text:00B13129                 lea     eax, [ebp+var_E38]
.text:00B1312F                 push    eax
.text:00B13130                 push    104h
.text:00B13135                 lea     eax, [ebp+var_20C]
.text:00B1313B                 push    eax
.text:00B1313C                 lea     eax, [esi-5]
.text:00B1313F                 push    eax
.text:00B13140                 call    sub_B11096
.text:00B13145                 add     esp, 18h
.text:00B13148                 test    edi, edi
.text:00B1314A                 jz      short loc_B1317E
.text:00B1314C                 push    ebx
.text:00B1314D                 push    [ebp+var_E30]
.text:00B13153                 lea     eax, [ebp+var_414]
.text:00B13159                 mov     ecx, edi
.text:00B1315B                 push    offset aRunTimeCheckFa ; "Run-Time Check Failure #%d - %s"
.text:00B13160                 push    eax
.text:00B13161                 push    [ebp+var_E38]
```

```
.text:00B13167                 lea     eax, [ebp+var_20C]
.text:00B1316D                 push    eax
.text:00B1316E                 push    [ebp+arg_4]
.text:00B13171                 call    ds:___guard_check_icall_fptr
.text:00B13177                 call    edi
.text:00B13179                 jmp     loc_B13214
.text:00B1317E ; ───────────────────────────────────────────────────────────────────────────
.text:00B1317E
.text:00B1317E loc_B1317E:                             ; CODE XREF: sub_B13030+11A↑j
.text:00B1317E                 push    0               ; lpUsedDefaultChar
.text:00B13180                 push    0               ; lpDefaultChar
.text:00B13182                 push    30Ah            ; cbMultiByte
.text:00B13187                 lea     eax, [ebp+MultiByteStr]
.text:00B1318D                 mov     esi, offset aUnknownFilenam ; "Unknown Filename"
.text:00B13192                 push    eax             ; lpMultiByteStr
.text:00B13193                 push    0FFFFFFFFh      ; cchWideChar
.text:00B13195                 lea     eax, [ebp+var_20C]
.text:00B1319B                 push    eax             ; lpWideCharStr
.text:00B1319C                 push    0               ; dwFlags
.text:00B1319E                 push    0FDE9h          ; CodePage
.text:00B131A3                 call    ds:WideCharToMultiByte
.text:00B131A9                 test    eax, eax
.text:00B131AB                 jz      short loc_B131B3
.text:00B131AD                 lea     esi, [ebp+MultiByteStr]
.text:00B131B3
.text:00B131B3 loc_B131B3:                             ; CODE XREF: sub_B13030+17B↑j
.text:00B131B3                 push    0               ; lpUsedDefaultChar
.text:00B131B5                 push    0               ; lpDefaultChar
.text:00B131B7                 push    30Ah            ; cbMultiByte
.text:00B131BC                 lea     eax, [ebp+var_A2C]
.text:00B131C2                 mov     edi, offset aUnknownModuleN ; "Unknown Module Name"
.text:00B131C7                 push    eax             ; lpMultiByteStr
.text:00B131C8                 push    0FFFFFFFFh      ; cchWideChar
.text:00B131CA                 lea     eax, [ebp+var_414]
.text:00B131D0                 push    eax             ; lpWideCharStr
.text:00B131D1                 push    0               ; dwFlags
.text:00B131D3                 push    0FDE9h          ; CodePage
.text:00B131D8                 call    ds:WideCharToMultiByte
.text:00B131DE                 test    eax, eax
.text:00B131E0                 jz      short loc_B131E8
.text:00B131E2                 lea     edi, [ebp+var_A2C]
.text:00B131E8
.text:00B131E8 loc_B131E8:                             ; CODE XREF: sub_B13030+1B0↑j
.text:00B131E8                 push    [ebp+var_E3C]
.text:00B131EE                 mov     ebx, [ebp+var_E34]
.text:00B131F4                 mov     ecx, ebx
.text:00B131F6                 push    [ebp+var_E30]
.text:00B131FC                 push    offset aRunTimeCheckFa_0 ; "Run-Time Check Failure #%d - %s"
.text:00B13201                 push    edi
.text:00B13202                 push    [ebp+var_E38]
.text:00B13208                 push    esi
.text:00B13209                 push    [ebp+arg_4]
.text:00B1320C                 call    ds:___guard_check_icall_fptr
.text:00B13212                 call    ebx
.text:00B13214
.text:00B13214 loc_B13214:                             ; CODE XREF: sub_B13030+149↑j
.text:00B13214                 add     esp, 1Ch
.text:00B13217                 cmp     eax, 1
.text:00B1321A                 jnz     short loc_B1321D
.text:00B1321C
.text:00B1321C loc_B1321C:                             ; CODE XREF: sub_B13030+D5↑j
.text:00B1321C                                         ; sub_B13030+E7↑j
.text:00B1321C                 int     3               ; Trap to Debugger
.text:00B1321D
.text:00B1321D loc_B1321D:                             ; CODE XREF: sub_B13030+C0↑j
.text:00B1321D                                         ; sub_B13030+1EA↑j
.text:00B1321D                 mov     ecx, [ebp+var_4]
.text:00B13220                 pop     edi
.text:00B13221                 pop     esi
.text:00B13222                 xor     ecx, ebp        ; StackCookie
.text:00B13224                 pop     ebx
.text:00B13225                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B1322A                 mov     esp, ebp
.text:00B1322C                 pop     ebp
.text:00B1322D                 retn
```

```
.text:00B1322D sub_B13030      endp
.text:00B1322D
.text:00B1322D ; ───────────────────────────────────────────────────────────────────────────
.text:00B1322E                 db 82h dup(0CCh)
.text:00B132B0
.text:00B132B0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B132B0
.text:00B132B0 ; Attributes: bp-based frame
.text:00B132B0
.text:00B132B0 ; int __cdecl sub_B132B0(ULONG_PTR *lpArguments)
.text:00B132B0 sub_B132B0      proc near               ; CODE XREF: sub_B12C60+21↑p
.text:00B132B0                                         ; sub_B12CB0+33↑p
.text:00B132B0
.text:00B132B0 ms_exc          = CPPEH_RECORD ptr -18h
.text:00B132B0 lpArguments     = dword ptr  8
.text:00B132B0
.text:00B132B0                 push    ebp
.text:00B132B1                 mov     ebp, esp
.text:00B132B3                 push    0FFFFFFFEh
.text:00B132B5                 push    offset stru_B1B2D8
.text:00B132BA                 push    offset SEH_4132B0
.text:00B132BF                 mov     eax, large fs:0
.text:00B132C5                 push    eax
.text:00B132C6                 sub     esp, 8
.text:00B132C9                 push    ebx
.text:00B132CA                 push    esi
.text:00B132CB                 push    edi
.text:00B132CC                 mov     eax, ___security_cookie
.text:00B132D1                 xor     [ebp+ms_exc.registration.ScopeTable], eax
.text:00B132D4                 xor     eax, ebp
.text:00B132D6                 push    eax
.text:00B132D7                 lea     eax, [ebp+ms_exc.registration]
.text:00B132DA                 mov     large fs:0, eax
.text:00B132E0                 mov     [ebp+ms_exc.old_esp], esp
.text:00B132E3                 mov     [ebp+ms_exc.registration.TryLevel], 0
.text:00B132EA                 push    [ebp+lpArguments] ; lpArguments
.text:00B132ED                 push    6               ; nNumberOfArguments
.text:00B132EF                 push    0               ; dwExceptionFlags
.text:00B132F1                 push    406D1388h       ; dwExceptionCode
.text:00B132F6                 call    ds:RaiseException
.text:00B132FC                 jmp     short loc_B13314
.text:00B132FE ; ───────────────────────────────────────────────────────────────────────────
.text:00B132FE
.text:00B132FE loc_B132FE:                             ; DATA XREF: .rdata:stru_B1B2D8↓o
.text:00B132FE                 mov     eax, [ebp+ms_exc.exc_ptr]
.text:00B13301                 mov     eax, [eax]
.text:00B13303                 xor     ecx, ecx
.text:00B13305                 cmp     dword ptr [eax], 406D1388h
.text:00B1330B                 setz    cl
.text:00B1330E                 mov     eax, ecx
.text:00B13310                 retn
.text:00B13311 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13311
.text:00B13311 loc_B13311:                             ; DATA XREF: .rdata:stru_B1B2D8↓o
.text:00B13311                 mov     esp, [ebp+ms_exc.old_esp]
.text:00B13314
.text:00B13314 loc_B13314:                             ; CODE XREF: sub_B132B0+4C↑j
.text:00B13314                 mov     [ebp+ms_exc.registration.TryLevel], 0FFFFFFFEh
.text:00B1331B                 mov     ecx, [ebp+ms_exc.registration.Next]
.text:00B1331E                 mov     large fs:0, ecx
.text:00B13325                 pop     ecx
.text:00B13326                 pop     edi
.text:00B13327                 pop     esi
.text:00B13328                 pop     ebx
.text:00B13329                 mov     esp, ebp
.text:00B1332B                 pop     ebp
.text:00B1332C                 retn
.text:00B1332C sub_B132B0      endp
.text:00B1332C
.text:00B1332C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1332D                 db 23h dup(0CCh)
.text:00B13350 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13350
.text:00B13350 loc_B13350:                             ; CODE XREF: .text:00B11366↑j
.text:00B13350                 push    ebp
```

```
.text:00B13351                 mov     ebp, esp
.text:00B13353                 sub     esp, 404h
.text:00B13359                 mov     eax, ___security_cookie
.text:00B1335E                 xor     eax, ebp
.text:00B13360                 mov     [ebp-4], eax
.text:00B13363                 push    esi
.text:00B13364                 mov     esi, [ebp+8]
.text:00B13367                 push    edi
.text:00B13368                 mov     edi, dword_B1C018
.text:00B1336E                 cmp     edi, 0FFFFFFFFh
.text:00B13371                 jz      short loc_B133E8
.text:00B13373                 test    esi, esi
.text:00B13375                 jz      short loc_B133D3
.text:00B13377                 push    esi
.text:00B13378                 call    sub_B13010
.text:00B1337D                 add     eax, 3Ah ; ':'
.text:00B13380                 add     esp, 4
.text:00B13383                 cmp     eax, 400h
.text:00B13388                 ja      short loc_B133D3
.text:00B1338A                 push    offset aTheVariable ; "The variable '"
.text:00B1338F                 lea     eax, [ebp-404h]
.text:00B13395                 push    400h
.text:00B1339A                 push    eax
.text:00B1339B                 call    j_strcpy_s
.text:00B133A0                 push    esi
.text:00B133A1                 lea     eax, [ebp-404h]
.text:00B133A7                 push    400h
.text:00B133AC                 push    eax
.text:00B133AD                 call    j_strcat_s
.text:00B133B2                 push    offset aIsBeingUsedWit ; "' is being used without being initializ"...
.text:00B133B7                 lea     eax, [ebp-404h]
.text:00B133BD                 push    400h
.text:00B133C2                 push    eax
.text:00B133C3                 call    j_strcat_s
.text:00B133C8                 add     esp, 24h
.text:00B133CB                 lea     eax, [ebp-404h]
.text:00B133D1                 jmp     short loc_B133D8
.text:00B133D3 ; ---------------------------------------------------------------------------
.text:00B133D3
.text:00B133D3 loc_B133D3:                             ; CODE XREF: .text:00B13375↑j
.text:00B133D3                                         ; .text:00B13388↑j
.text:00B133D3                 mov     eax, offset aAVariableIsBei ; "A variable is being used without being "...
.text:00B133D8
.text:00B133D8 loc_B133D8:                             ; CODE XREF: .text:00B133D1↑j
.text:00B133D8                 push    eax
.text:00B133D9                 mov     eax, [ebp+4]
.text:00B133DC                 push    3
.text:00B133DE                 push    edi
.text:00B133DF                 push    eax
.text:00B133E0                 call    sub_B13030
.text:00B133E5                 add     esp, 10h
.text:00B133E8
.text:00B133E8 loc_B133E8:                             ; CODE XREF: .text:00B13371↑j
.text:00B133E8                 mov     ecx, [ebp-4]
.text:00B133EB                 pop     edi
.text:00B133EC                 xor     ecx, ebp
.text:00B133EE                 pop     esi
.text:00B133EF                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B133F4                 mov     esp, ebp
.text:00B133F6                 pop     ebp
.text:00B133F7                 retn
.text:00B133F7 ; ---------------------------------------------------------------------------
.text:00B133F8                 db 38h dup(0CCh)
.text:00B13430
.text:00B13430 ; =============== S U B R O U T I N E =======================================
.text:00B13430
.text:00B13430
.text:00B13430 sub_B13430      proc near               ; CODE XREF: sub_B1135C↑j
.text:00B13430                 mov     eax, offset unk_B1C140
.text:00B13435                 retn
.text:00B13435 sub_B13430      endp
.text:00B13435
.text:00B13435 ; ---------------------------------------------------------------------------
.text:00B13436                 align 10h
.text:00B13440
```

```
.text:00B13440 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13440
.text:00B13440 ; Attributes: bp-based frame
.text:00B13440
.text:00B13440 ; int __cdecl sub_B13440(char *Buffer, size_t BufferCount, char *Format, _locale_t Locale, va_list
ArgList)
.text:00B13440 sub_B13440      proc near               ; CODE XREF: sub_B1120D↑j
.text:00B13440
.text:00B13440 Buffer          = dword ptr  8
.text:00B13440 BufferCount     = dword ptr  0Ch
.text:00B13440 Format          = dword ptr  10h
.text:00B13440 Locale          = dword ptr  14h
.text:00B13440 ArgList         = dword ptr  18h
.text:00B13440
.text:00B13440                 push    ebp
.text:00B13441                 mov     ebp, esp
.text:00B13443                 call    sub_B1135C
.text:00B13448                 push    [ebp+ArgList]   ; ArgList
.text:00B1344B                 push    [ebp+Locale]    ; Locale
.text:00B1344E                 push    [ebp+Format]    ; Format
.text:00B13451                 push    [ebp+BufferCount] ; BufferCount
.text:00B13454                 push    [ebp+Buffer]    ; Buffer
.text:00B13457                 push    dword ptr [eax+4]
.text:00B1345A                 push    dword ptr [eax] ; Options
.text:00B1345C                 call    j___stdio_common_vsprintf_s
.text:00B13461                 add     esp, 1Ch
.text:00B13464                 test    eax, eax
.text:00B13466                 jns     short loc_B1346B
.text:00B13468                 or      eax, 0FFFFFFFFh
.text:00B1346B
.text:00B1346B loc_B1346B:                             ; CODE XREF: sub_B13440+26↑j
.text:00B1346B                 pop     ebp
.text:00B1346C                 retn
.text:00B1346C sub_B13440      endp
.text:00B1346C
.text:00B1346C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1346D                 align 20h
.text:00B13480
.text:00B13480 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13480
.text:00B13480 ; Attributes: bp-based frame
.text:00B13480
.text:00B13480 ; int __cdecl sub_B13480(char *Buffer, size_t BufferCount, char *Format, char ArgList)
.text:00B13480 sub_B13480      proc near               ; CODE XREF: sub_B11334↑j
.text:00B13480
.text:00B13480 Buffer          = dword ptr  8
.text:00B13480 BufferCount     = dword ptr  0Ch
.text:00B13480 Format          = dword ptr  10h
.text:00B13480 ArgList         = byte ptr  14h
.text:00B13480
.text:00B13480                 push    ebp
.text:00B13481                 mov     ebp, esp
.text:00B13483                 lea     eax, [ebp+ArgList]
.text:00B13486                 push    eax             ; ArgList
.text:00B13487                 push    0               ; Locale
.text:00B13489                 push    [ebp+Format]    ; Format
.text:00B1348C                 push    [ebp+BufferCount] ; BufferCount
.text:00B1348F                 push    [ebp+Buffer]    ; Buffer
.text:00B13492                 call    sub_B1120D
.text:00B13497                 add     esp, 14h
.text:00B1349A                 pop     ebp
.text:00B1349B                 retn
.text:00B1349B sub_B13480      endp
.text:00B1349B
.text:00B1349B ; ───────────────────────────────────────────────────────────────────────────
.text:00B1349C                 db 14h dup(0CCh)
.text:00B134B0
.text:00B134B0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B134B0
.text:00B134B0 ; Attributes: bp-based frame
.text:00B134B0
.text:00B134B0 ; int __cdecl sub_B134B0(struct _EXCEPTION_POINTERS *ExceptionInfo)
.text:00B134B0 sub_B134B0      proc near               ; CODE XREF: sub_B11429↑j
.text:00B134B0
.text:00B134B0 ExceptionInfo   = dword ptr  8
```

```
.text:00B134B0
.text:00B134B0                 push    ebp
.text:00B134B1                 mov     ebp, esp
.text:00B134B3                 push    0               ; lpTopLevelExceptionFilter
.text:00B134B5                 call    ds:SetUnhandledExceptionFilter
.text:00B134BB                 mov     eax, [ebp+ExceptionInfo]
.text:00B134BE                 push    eax             ; ExceptionInfo
.text:00B134BF                 call    ds:UnhandledExceptionFilter
.text:00B134C5                 push    0C0000409h      ; uExitCode
.text:00B134CA                 call    ds:GetCurrentProcess
.text:00B134D0                 push    eax             ; hProcess
.text:00B134D1                 call    ds:TerminateProcess
.text:00B134D7                 pop     ebp
.text:00B134D8                 retn
.text:00B134D8 sub_B134B0      endp
.text:00B134D8
.text:00B134D8 ; ───────────────────────────────────────────────────────────────────────────
.text:00B134D9                 db 17h dup(0CCh)
.text:00B134F0
.text:00B134F0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B134F0
.text:00B134F0 ; Attributes: bp-based frame
.text:00B134F0
.text:00B134F0 sub_B134F0      proc near               ; CODE XREF: sub_B1102D↑j
.text:00B134F0
.text:00B134F0 var_324         = dword ptr -324h
.text:00B134F0 var_8           = dword ptr -8
.text:00B134F0 var_s0          = dword ptr  0
.text:00B134F0 arg_0           = byte ptr  8
.text:00B134F0
.text:00B134F0                 push    ebp
.text:00B134F1                 mov     ebp, esp
.text:00B134F3                 sub     esp, 324h
.text:00B134F9                 push    17h             ; ProcessorFeature
.text:00B134FB                 call    ds:IsProcessorFeaturePresent
.text:00B13501                 test    eax, eax
.text:00B13503                 jz      short loc_B1350C
.text:00B13505                 mov     ecx, 2
.text:00B1350A                 int     29h             ; Win8: RtlFailFast(ecx)
.text:00B1350C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1350C
.text:00B1350C loc_B1350C:                             ; CODE XREF: sub_B134F0+13↑j
.text:00B1350C                 mov     dword_B1C250, eax
.text:00B13511                 mov     dword_B1C24C, ecx
.text:00B13517                 mov     dword_B1C248, edx
.text:00B1351D                 mov     dword_B1C244, ebx
.text:00B13523                 mov     dword_B1C240, esi
.text:00B13529                 mov     dword_B1C23C, edi
.text:00B1352F                 mov     word_B1C268, ss
.text:00B13536                 mov     word_B1C25C, cs
.text:00B1353D                 mov     word_B1C238, ds
.text:00B13544                 mov     word_B1C234, es
.text:00B1354B                 mov     word_B1C230, fs
.text:00B13552                 mov     word_B1C22C, gs
.text:00B13559                 pushf
.text:00B1355A                 pop     dword_B1C260
.text:00B13560                 mov     eax, [ebp+var_s0]
.text:00B13563                 mov     dword_B1C254, eax
.text:00B13568                 mov     eax, [ebp+4]
.text:00B1356B                 mov     dword_B1C258, eax
.text:00B13570                 lea     eax, [ebp+arg_0]
.text:00B13573                 mov     dword_B1C264, eax
.text:00B13578                 mov     eax, [ebp+var_324]
.text:00B1357E                 mov     dword_B1C1A0, 10001h
.text:00B13588                 mov     eax, dword_B1C258
.text:00B1358D                 mov     dword_B1C15C, eax
.text:00B13592                 mov     dword_B1C150, 0C0000409h
.text:00B1359C                 mov     dword_B1C154, 1
.text:00B135A6                 mov     dword_B1C160, 1
.text:00B135B0                 mov     ecx, 4
.text:00B135B5                 imul    edx, ecx, 0
.text:00B135B8                 mov     dword_B1C164[edx], 2
.text:00B135C2                 mov     eax, 4
.text:00B135C7                 imul    ecx, eax, 0
.text:00B135CA                 mov     edx, ___security_cookie
```

```
.text:00B135D0                 mov     [ebp+ecx+var_8], edx
.text:00B135D4                 mov     eax, 4
.text:00B135D9                 shl     eax, 0
.text:00B135DC                 mov     ecx, dword_B1C000
.text:00B135E2                 mov     [ebp+eax+var_8], ecx
.text:00B135E6                 push    offset ExceptionInfo ; ExceptionInfo
.text:00B135EB                 call    sub_B11429
.text:00B135F0                 mov     esp, ebp
.text:00B135F2                 pop     ebp
.text:00B135F3                 retn
.text:00B135F3 sub_B134F0      endp
.text:00B135F3
.text:00B135F3 ; ───────────────────────────────────────────────────────────────────────────
.text:00B135F4                 db 4Ch dup(0CCh)
.text:00B13640 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13640                 push    ebp
.text:00B13641                 mov     ebp, esp
.text:00B13643                 push    8
.text:00B13645                 call    sub_B113D4
.text:00B1364A                 pop     ebp
.text:00B1364B                 retn
.text:00B1364B ; ───────────────────────────────────────────────────────────────────────────
.text:00B1364C                 align 10h
.text:00B13650
.text:00B13650 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13650
.text:00B13650 ; Attributes: bp-based frame
.text:00B13650
.text:00B13650 sub_B13650      proc near               ; CODE XREF: sub_B113D4↑j
.text:00B13650
.text:00B13650 var_31C         = dword ptr -31Ch
.text:00B13650 var_s0          = dword ptr  0
.text:00B13650 arg_0           = dword ptr  8
.text:00B13650
.text:00B13650                 push    ebp
.text:00B13651                 mov     ebp, esp
.text:00B13653                 sub     esp, 31Ch
.text:00B13659                 push    17h             ; ProcessorFeature
.text:00B1365B                 call    ds:IsProcessorFeaturePresent
.text:00B13661                 test    eax, eax
.text:00B13663                 jz      short loc_B1366A
.text:00B13665                 mov     ecx, [ebp+arg_0]
.text:00B13668                 int     29h             ; Win8: RtlFailFast(ecx)
.text:00B1366A ; ───────────────────────────────────────────────────────────────────────────
.text:00B1366A
.text:00B1366A loc_B1366A:                             ; CODE XREF: sub_B13650+13↑j
.text:00B1366A                 mov     dword_B1C250, eax
.text:00B1366F                 mov     dword_B1C24C, ecx
.text:00B13675                 mov     dword_B1C248, edx
.text:00B1367B                 mov     dword_B1C244, ebx
.text:00B13681                 mov     dword_B1C240, esi
.text:00B13687                 mov     dword_B1C23C, edi
.text:00B1368D                 mov     word_B1C268, ss
.text:00B13694                 mov     word_B1C25C, cs
.text:00B1369B                 mov     word_B1C238, ds
.text:00B136A2                 mov     word_B1C234, es
.text:00B136A9                 mov     word_B1C230, fs
.text:00B136B0                 mov     word_B1C22C, gs
.text:00B136B7                 pushf
.text:00B136B8                 pop     dword_B1C260
.text:00B136BE                 mov     eax, [ebp+var_s0]
.text:00B136C1                 mov     dword_B1C254, eax
.text:00B136C6                 mov     eax, [ebp+4]
.text:00B136C9                 mov     dword_B1C258, eax
.text:00B136CE                 lea     eax, [ebp+arg_0]
.text:00B136D1                 mov     dword_B1C264, eax
.text:00B136D6                 mov     eax, [ebp+var_31C]
.text:00B136DC                 mov     eax, dword_B1C258
.text:00B136E1                 mov     dword_B1C15C, eax
.text:00B136E6                 mov     dword_B1C150, 0C0000409h
.text:00B136F0                 mov     dword_B1C154, 1
.text:00B136FA                 mov     dword_B1C160, 1
.text:00B13704                 mov     ecx, 4
.text:00B13709                 imul    edx, ecx, 0
.text:00B1370C                 mov     eax, [ebp+arg_0]
```

```
.text:00B1370F                 mov     dword_B1C164[edx], eax
.text:00B13715                 push    offset ExceptionInfo ; ExceptionInfo
.text:00B1371A                 call    sub_B11429
.text:00B1371F                 mov     esp, ebp
.text:00B13721                 pop     ebp
.text:00B13722                 retn
.text:00B13722 sub_B13650      endp
.text:00B13722
.text:00B13722 ; ---------------------------------------------------------------------------
.text:00B13723                 db 3Dh dup(0CCh)
.text:00B13760 ; ---------------------------------------------------------------------------
.text:00B13760
.text:00B13760 loc_B13760:                             ; CODE XREF: .text:00B111F4↑j
.text:00B13760                 push    ebp
.text:00B13761                 mov     ebp, esp
.text:00B13763                 sub     esp, 320h
.text:00B13769                 push    17h
.text:00B1376B                 call    ds:IsProcessorFeaturePresent
.text:00B13771                 test    eax, eax
.text:00B13773                 jz      short loc_B1377A
.text:00B13775                 mov     ecx, [ebp+8]
.text:00B13778                 int     29h             ; Win8: RtlFailFast(ecx)
.text:00B1377A ; ---------------------------------------------------------------------------
.text:00B1377A
.text:00B1377A loc_B1377A:                             ; CODE XREF: .text:00B13773↑j
.text:00B1377A                 mov     dword_B1C250, eax
.text:00B1377F                 mov     dword_B1C24C, ecx
.text:00B13785                 mov     dword_B1C248, edx
.text:00B1378B                 mov     dword_B1C244, ebx
.text:00B13791                 mov     dword_B1C240, esi
.text:00B13797                 mov     dword_B1C23C, edi
.text:00B1379D                 mov     word_B1C268, ss
.text:00B137A4                 mov     word_B1C25C, cs
.text:00B137AB                 mov     word_B1C238, ds
.text:00B137B2                 mov     word_B1C234, es
.text:00B137B9                 mov     word_B1C230, fs
.text:00B137C0                 mov     word_B1C22C, gs
.text:00B137C7                 pushf
.text:00B137C8                 pop     dword_B1C260
.text:00B137CE                 mov     eax, [ebp+0]
.text:00B137D1                 mov     dword_B1C254, eax
.text:00B137D6                 mov     eax, [ebp+4]
.text:00B137D9                 mov     dword_B1C258, eax
.text:00B137DE                 lea     eax, [ebp+8]
.text:00B137E1                 mov     dword_B1C264, eax
.text:00B137E6                 mov     eax, [ebp-320h]
.text:00B137EC                 mov     eax, dword_B1C258
.text:00B137F1                 mov     dword_B1C15C, eax
.text:00B137F6                 mov     dword_B1C150, 0C0000409h
.text:00B13800                 mov     dword_B1C154, 1
.text:00B1380A                 cmp     dword ptr [ebp+0Ch], 0
.text:00B1380E                 jbe     short loc_B1381D
.text:00B13810                 cmp     dword ptr [ebp+10h], 0
.text:00B13814                 jnz     short loc_B1381D
.text:00B13816                 mov     dword ptr [ebp+0Ch], 0
.text:00B1381D
.text:00B1381D loc_B1381D:                             ; CODE XREF: .text:00B1380E↑j
.text:00B1381D                                         ; .text:00B13814↑j
.text:00B1381D                 cmp     dword ptr [ebp+0Ch], 0Eh
.text:00B13821                 jbe     short loc_B1382C
.text:00B13823                 mov     ecx, [ebp+0Ch]
.text:00B13826                 sub     ecx, 1
.text:00B13829                 mov     [ebp+0Ch], ecx
.text:00B1382C
.text:00B1382C loc_B1382C:                             ; CODE XREF: .text:00B13821↑j
.text:00B1382C                 mov     edx, [ebp+0Ch]
.text:00B1382F                 add     edx, 1
.text:00B13832                 mov     dword_B1C160, edx
.text:00B13838                 mov     eax, 4
.text:00B1383D                 imul    ecx, eax, 0
.text:00B13840                 mov     edx, [ebp+8]
.text:00B13843                 mov     dword_B1C164[ecx], edx
.text:00B13849                 mov     dword ptr [ebp-4], 0
.text:00B13850                 jmp     short loc_B1385B
.text:00B13852 ; ---------------------------------------------------------------------------
```

```
.text:00B13852
.text:00B13852 loc_B13852:                              ; CODE XREF: .text:00B13876↓j
.text:00B13852                 mov     eax, [ebp-4]
.text:00B13855                 add     eax, 1
.text:00B13858                 mov     [ebp-4], eax
.text:00B1385B
.text:00B1385B loc_B1385B:                              ; CODE XREF: .text:00B13850↑j
.text:00B1385B                 mov     ecx, [ebp-4]
.text:00B1385E                 cmp     ecx, [ebp+0Ch]
.text:00B13861                 jnb     short loc_B13878
.text:00B13863                 mov     edx, [ebp-4]
.text:00B13866                 mov     eax, [ebp-4]
.text:00B13869                 mov     ecx, [ebp+10h]
.text:00B1386C                 mov     eax, [ecx+eax*4]
.text:00B1386F                 mov     dword_B1C168[edx*4], eax
.text:00B13876                 jmp     short loc_B13852
.text:00B13878 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13878
.text:00B13878 loc_B13878:                              ; CODE XREF: .text:00B13861↑j
.text:00B13878                 push    offset ExceptionInfo
.text:00B1387D                 call    sub_B11429
.text:00B13882                 mov     esp, ebp
.text:00B13884                 pop     ebp
.text:00B13885                 retn
.text:00B13885 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13886                 db 4Ah dup(0CCh)
.text:00B138D0
.text:00B138D0 ; =============== S U B R O U T I N E =======================================
.text:00B138D0
.text:00B138D0
.text:00B138D0 sub_B138D0      proc near                ; CODE XREF: sub_B110A0↑j
.text:00B138D0                 mov     eax, dword_B1C50C
.text:00B138D5                 retn
.text:00B138D5 sub_B138D0      endp
.text:00B138D5
.text:00B138D5 ; ───────────────────────────────────────────────────────────────────────────
.text:00B138D6                 align 10h
.text:00B138E0
.text:00B138E0 ; =============== S U B R O U T I N E =======================================
.text:00B138E0
.text:00B138E0
.text:00B138E0 sub_B138E0      proc near                ; CODE XREF: sub_B113F2↑j
.text:00B138E0                 mov     eax, dword_B1C510
.text:00B138E5                 retn
.text:00B138E5 sub_B138E0      endp
.text:00B138E5
.text:00B138E5 ; ───────────────────────────────────────────────────────────────────────────
.text:00B138E6                 align 10h
.text:00B138F0
.text:00B138F0 loc_B138F0:                              ; CODE XREF: .text:00B1118B↑j
.text:00B138F0                 push    ebp
.text:00B138F1                 mov     ebp, esp
.text:00B138F3                 mov     eax, [ebp+8]
.text:00B138F6                 cmp     eax, 4
.text:00B138F9                 ja      short loc_B13904
.text:00B138FB                 mov     eax, ds:off_B1A20C[eax*4] ; "Stack pointer corruption"
.text:00B13902                 pop     ebp
.text:00B13903                 retn
.text:00B13904 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13904
.text:00B13904 loc_B13904:                              ; CODE XREF: .text:00B138F9↑j
.text:00B13904                 xor     eax, eax
.text:00B13906                 pop     ebp
.text:00B13907                 retn
.text:00B13907 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13908                 align 10h
.text:00B13910
.text:00B13910 loc_B13910:                              ; CODE XREF: .text:00B11032↑j
.text:00B13910                 mov     eax, 5
.text:00B13915                 retn
.text:00B13915 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13916                 align 10h
.text:00B13920
.text:00B13920 loc_B13920:                              ; CODE XREF: .text:00B111D6↑j
.text:00B13920                 push    ebp
```

```
.text:00B13921                 mov     ebp, esp
.text:00B13923                 mov     eax, dword_B1C50C
.text:00B13928                 mov     ecx, [ebp+8]
.text:00B1392B                 mov     dword_B1C50C, ecx
.text:00B13931                 mov     dword_B1C510, 0
.text:00B1393B                 pop     ebp
.text:00B1393C                 retn
.text:00B1393C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1393D                 db 13h dup(0CCh)
.text:00B13950
.text:00B13950 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13950
.text:00B13950 ; Attributes: bp-based frame
.text:00B13950
.text:00B13950 sub_B13950      proc near               ; CODE XREF: sub_B11393↑j
.text:00B13950
.text:00B13950 arg_0           = dword ptr  8
.text:00B13950
.text:00B13950                 push    ebp
.text:00B13951                 mov     ebp, esp
.text:00B13953                 mov     eax, dword_B1C510
.text:00B13958                 mov     ecx, [ebp+arg_0]
.text:00B1395B                 mov     dword_B1C510, ecx
.text:00B13961                 mov     dword_B1C50C, 0
.text:00B1396B                 pop     ebp
.text:00B1396C                 retn
.text:00B1396C sub_B13950      endp
.text:00B1396C
.text:00B1396C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1396D                 align 20h
.text:00B13980
.text:00B13980 loc_B13980:                             ; CODE XREF: .text:00B11190↑j
.text:00B13980                 push    ebp
.text:00B13981                 mov     ebp, esp
.text:00B13983                 mov     edx, [ebp+8]
.text:00B13986                 cmp     edx, 4
.text:00B13989                 ja      short loc_B1399E
.text:00B1398B                 mov     eax, dword_B1C00C[edx*4]
.text:00B13992                 mov     ecx, [ebp+0Ch]
.text:00B13995                 mov     dword_B1C00C[edx*4], ecx
.text:00B1399C                 pop     ebp
.text:00B1399D                 retn
.text:00B1399E ; ───────────────────────────────────────────────────────────────────────────
.text:00B1399E
.text:00B1399E loc_B1399E:                             ; CODE XREF: .text:00B13989↑j
.text:00B1399E                 or      eax, 0FFFFFFFFh
.text:00B139A1                 pop     ebp
.text:00B139A2                 retn
.text:00B139A2 ; ───────────────────────────────────────────────────────────────────────────
.text:00B139A3                 align 10h
.text:00B139B0
.text:00B139B0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B139B0
.text:00B139B0 ; Attributes: bp-based frame
.text:00B139B0
.text:00B139B0 sub_B139B0      proc near               ; CODE XREF: sub_B13E60+7E↓p
.text:00B139B0
.text:00B139B0 var_18          = dword ptr -18h
.text:00B139B0 var_14          = dword ptr -14h
.text:00B139B0 var_10          = dword ptr -10h
.text:00B139B0 var_C           = dword ptr -0Ch
.text:00B139B0 var_8           = dword ptr -8
.text:00B139B0 var_4           = dword ptr -4
.text:00B139B0 arg_0           = dword ptr  8
.text:00B139B0 arg_4           = dword ptr  0Ch
.text:00B139B0
.text:00B139B0                 push    ebp
.text:00B139B1                 mov     ebp, esp
.text:00B139B3                 sub     esp, 18h
.text:00B139B6                 mov     eax, [ebp+arg_0]
.text:00B139B9                 mov     [ebp+var_C], eax
.text:00B139BC                 mov     ecx, [ebp+var_C]
.text:00B139BF                 mov     edx, [ebp+var_C]
.text:00B139C2                 add     edx, [ecx+3Ch]
.text:00B139C5                 mov     [ebp+var_14], edx
```

```
.text:00B139C8                 mov     eax, [ebp+var_14]
.text:00B139CB                 mov     [ebp+var_8], eax
.text:00B139CE                 mov     ecx, [ebp+var_8]
.text:00B139D1                 movzx   edx, word ptr [ecx+14h]
.text:00B139D5                 mov     eax, [ebp+var_8]
.text:00B139D8                 lea     ecx, [eax+edx+18h]
.text:00B139DC                 mov     [ebp+var_10], ecx
.text:00B139DF                 mov     edx, [ebp+var_8]
.text:00B139E2                 movzx   eax, word ptr [edx+6]
.text:00B139E6                 imul    ecx, eax, 28h ; '('
.text:00B139E9                 add     ecx, [ebp+var_10]
.text:00B139EC                 mov     [ebp+var_18], ecx
.text:00B139EF                 mov     edx, [ebp+var_10]
.text:00B139F2                 mov     [ebp+var_4], edx
.text:00B139F5                 jmp     short loc_B13A00
.text:00B139F7 ; ───────────────────────────────────────────────────────────────────────────
.text:00B139F7
.text:00B139F7 loc_B139F7:                             ; CODE XREF: sub_B139B0:loc_B13A29↓j
.text:00B139F7                 mov     eax, [ebp+var_4]
.text:00B139FA                 add     eax, 28h ; '('
.text:00B139FD                 mov     [ebp+var_4], eax
.text:00B13A00
.text:00B13A00 loc_B13A00:                             ; CODE XREF: sub_B139B0+45↑j
.text:00B13A00                 mov     ecx, [ebp+var_4]
.text:00B13A03                 cmp     ecx, [ebp+var_18]
.text:00B13A06                 jz      short loc_B13A2B
.text:00B13A08                 mov     edx, [ebp+var_4]
.text:00B13A0B                 mov     eax, [ebp+arg_4]
.text:00B13A0E                 cmp     eax, [edx+0Ch]
.text:00B13A11                 jb      short loc_B13A29
.text:00B13A13                 mov     ecx, [ebp+var_4]
.text:00B13A16                 mov     edx, [ecx+0Ch]
.text:00B13A19                 mov     eax, [ebp+var_4]
.text:00B13A1C                 add     edx, [eax+8]
.text:00B13A1F                 cmp     [ebp+arg_4], edx
.text:00B13A22                 jnb     short loc_B13A29
.text:00B13A24                 mov     eax, [ebp+var_4]
.text:00B13A27                 jmp     short loc_B13A2D
.text:00B13A29 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13A29
.text:00B13A29 loc_B13A29:                             ; CODE XREF: sub_B139B0+61↑j
.text:00B13A29                                         ; sub_B139B0+72↑j
.text:00B13A29                 jmp     short loc_B139F7
.text:00B13A2B ; ───────────────────────────────────────────────────────────────────────────
.text:00B13A2B
.text:00B13A2B loc_B13A2B:                             ; CODE XREF: sub_B139B0+56↑j
.text:00B13A2B                 xor     eax, eax
.text:00B13A2D
.text:00B13A2D loc_B13A2D:                             ; CODE XREF: sub_B139B0+77↑j
.text:00B13A2D                 mov     esp, ebp
.text:00B13A2F                 pop     ebp
.text:00B13A30                 retn
.text:00B13A30 sub_B139B0      endp
.text:00B13A30
.text:00B13A30 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13A31                 db 2Fh dup(0CCh)
.text:00B13A60
.text:00B13A60 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13A60
.text:00B13A60 ; Attributes: bp-based frame
.text:00B13A60
.text:00B13A60 sub_B13A60      proc near               ; CODE XREF: sub_B13E60+4B↓p
.text:00B13A60
.text:00B13A60 var_10          = dword ptr -10h
.text:00B13A60 var_C           = dword ptr -0Ch
.text:00B13A60 var_8           = dword ptr -8
.text:00B13A60 var_4           = dword ptr -4
.text:00B13A60 arg_0           = dword ptr  8
.text:00B13A60
.text:00B13A60                 push    ebp
.text:00B13A61                 mov     ebp, esp
.text:00B13A63                 sub     esp, 10h
.text:00B13A66                 cmp     [ebp+arg_0], 0
.text:00B13A6A                 jnz     short loc_B13A70
.text:00B13A6C                 xor     al, al
```

```
.text:00B13A6E                 jmp     short loc_B13AC5
.text:00B13A70 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13A70
.text:00B13A70 loc_B13A70:                             ; CODE XREF: sub_B13A60+A↑j
.text:00B13A70                 mov     eax, [ebp+arg_0]
.text:00B13A73                 mov     [ebp+var_4], eax
.text:00B13A76                 mov     ecx, [ebp+var_4]
.text:00B13A79                 movzx   edx, word ptr [ecx]
.text:00B13A7C                 cmp     edx, 5A4Dh
.text:00B13A82                 jz      short loc_B13A88
.text:00B13A84                 xor     al, al
.text:00B13A86                 jmp     short loc_B13AC5
.text:00B13A88 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13A88
.text:00B13A88 loc_B13A88:                             ; CODE XREF: sub_B13A60+22↑j
.text:00B13A88                 mov     eax, [ebp+var_4]
.text:00B13A8B                 mov     ecx, [ebp+var_4]
.text:00B13A8E                 add     ecx, [eax+3Ch]
.text:00B13A91                 mov     [ebp+var_C], ecx
.text:00B13A94                 mov     edx, [ebp+var_C]
.text:00B13A97                 mov     [ebp+var_8], edx
.text:00B13A9A                 mov     eax, [ebp+var_8]
.text:00B13A9D                 cmp     dword ptr [eax], 4550h
.text:00B13AA3                 jz      short loc_B13AA9
.text:00B13AA5                 xor     al, al
.text:00B13AA7                 jmp     short loc_B13AC5
.text:00B13AA9 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13AA9
.text:00B13AA9 loc_B13AA9:                             ; CODE XREF: sub_B13A60+43↑j
.text:00B13AA9                 mov     ecx, [ebp+var_8]
.text:00B13AAC                 add     ecx, 18h
.text:00B13AAF                 mov     [ebp+var_10], ecx
.text:00B13AB2                 mov     edx, [ebp+var_10]
.text:00B13AB5                 movzx   eax, word ptr [edx]
.text:00B13AB8                 cmp     eax, 10Bh
.text:00B13ABD                 jz      short loc_B13AC3
.text:00B13ABF                 xor     al, al
.text:00B13AC1                 jmp     short loc_B13AC5
.text:00B13AC3 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13AC3
.text:00B13AC3 loc_B13AC3:                             ; CODE XREF: sub_B13A60+5D↑j
.text:00B13AC3                 mov     al, 1
.text:00B13AC5
.text:00B13AC5 loc_B13AC5:                             ; CODE XREF: sub_B13A60+E↑j
.text:00B13AC5                                         ; sub_B13A60+26↑j ...
.text:00B13AC5                 mov     esp, ebp
.text:00B13AC7                 pop     ebp
.text:00B13AC8                 retn
.text:00B13AC8 sub_B13A60      endp
.text:00B13AC8
.text:00B13AC8 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13AC9                 db 27h dup(0CCh)
.text:00B13AF0
.text:00B13AF0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13AF0
.text:00B13AF0 ; Attributes: bp-based frame
.text:00B13AF0
.text:00B13AF0 sub_B13AF0      proc near               ; CODE XREF: sub_B113E8↑j
.text:00B13AF0                 push    ebp
.text:00B13AF1                 mov     ebp, esp
.text:00B13AF3                 mov     eax, large fs:18h
.text:00B13AF9                 pop     ebp
.text:00B13AFA                 retn
.text:00B13AFA sub_B13AF0      endp
.text:00B13AFA
.text:00B13AFA ; ───────────────────────────────────────────────────────────────────────────
.text:00B13AFB                 align 10h
.text:00B13B00
.text:00B13B00 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13B00
.text:00B13B00 ; Attributes: bp-based frame
.text:00B13B00
.text:00B13B00 sub_B13B00      proc near               ; CODE XREF: sub_B11343↑j
.text:00B13B00
.text:00B13B00 var_8           = dword ptr -8
```

```
.text:00B13B00 var_4           = dword ptr -4
.text:00B13B00
.text:00B13B00                 push    ebp
.text:00B13B01                 mov     ebp, esp
.text:00B13B03                 sub     esp, 8
.text:00B13B06                 call    sub_B111E5
.text:00B13B0B                 test    eax, eax
.text:00B13B0D                 jnz     short loc_B13B13
.text:00B13B0F                 xor     al, al
.text:00B13B11                 jmp     short loc_B13B45
.text:00B13B13 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13B13
.text:00B13B13 loc_B13B13:                             ; CODE XREF: sub_B13B00+D↑j
.text:00B13B13                 call    sub_B113E8
.text:00B13B18                 mov     eax, [eax+4]
.text:00B13B1B                 mov     [ebp+var_4], eax
.text:00B13B1E
.text:00B13B1E loc_B13B1E:                             ; CODE XREF: sub_B13B00:loc_B13B41↓j
.text:00B13B1E                 mov     ecx, [ebp+var_4]
.text:00B13B21                 mov     edx, offset unk_B1C51C
.text:00B13B26                 xor     eax, eax
.text:00B13B28                 lock cmpxchg [edx], ecx
.text:00B13B2C                 mov     [ebp+var_8], eax
.text:00B13B2F                 cmp     [ebp+var_8], 0
.text:00B13B33                 jz      short loc_B13B43
.text:00B13B35                 mov     eax, [ebp+var_4]
.text:00B13B38                 cmp     eax, [ebp+var_8]
.text:00B13B3B                 jnz     short loc_B13B41
.text:00B13B3D                 mov     al, 1
.text:00B13B3F                 jmp     short loc_B13B45
.text:00B13B41 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13B41
.text:00B13B41 loc_B13B41:                             ; CODE XREF: sub_B13B00+3B↑j
.text:00B13B41                 jmp     short loc_B13B1E
.text:00B13B43 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13B43
.text:00B13B43 loc_B13B43:                             ; CODE XREF: sub_B13B00+33↑j
.text:00B13B43                 xor     al, al
.text:00B13B45
.text:00B13B45 loc_B13B45:                             ; CODE XREF: sub_B13B00+11↑j
.text:00B13B45                                         ; sub_B13B00+3F↑j
.text:00B13B45                 mov     esp, ebp
.text:00B13B47                 pop     ebp
.text:00B13B48                 retn
.text:00B13B48 sub_B13B00      endp
.text:00B13B48
.text:00B13B48 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13B49                 align 20h
.text:00B13B60
.text:00B13B60 loc_B13B60:                             ; CODE XREF: .text:00B11172↑j
.text:00B13B60                 push    ebp
.text:00B13B61                 mov     ebp, esp
.text:00B13B63                 call    sub_B111E5
.text:00B13B68                 test    eax, eax
.text:00B13B6A                 jz      short loc_B13B73
.text:00B13B6C                 call    sub_B111EA
.text:00B13B71                 jmp     short loc_B13B85
.text:00B13B73 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13B73
.text:00B13B73 loc_B13B73:                             ; CODE XREF: .text:00B13B6A↑j
.text:00B13B73                 call    sub_B11258
.text:00B13B78                 test    eax, eax
.text:00B13B7A                 jz      short loc_B13B80
.text:00B13B7C                 xor     al, al
.text:00B13B7E                 jmp     short loc_B13B87
.text:00B13B80 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13B80
.text:00B13B80 loc_B13B80:                             ; CODE XREF: .text:00B13B7A↑j
.text:00B13B80                 call    sub_B11163
.text:00B13B85
.text:00B13B85 loc_B13B85:                             ; CODE XREF: .text:00B13B71↑j
.text:00B13B85                 mov     al, 1
.text:00B13B87
.text:00B13B87 loc_B13B87:                             ; CODE XREF: .text:00B13B7E↑j
.text:00B13B87                 pop     ebp
```

```
.text:00B13B88                 retn
.text:00B13B88 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13B89                 align 20h
.text:00B13BA0
.text:00B13BA0 loc_B13BA0:                             ; CODE XREF: .text:00B110D7↑j
.text:00B13BA0                 push    ebp
.text:00B13BA1                 mov     ebp, esp
.text:00B13BA3                 push    0
.text:00B13BA5                 call    sub_B11168
.text:00B13BAA                 add     esp, 4
.text:00B13BAD                 movzx   eax, al
.text:00B13BB0                 test    eax, eax
.text:00B13BB2                 jnz     short loc_B13BB8
.text:00B13BB4                 xor     al, al
.text:00B13BB6                 jmp     short loc_B13BBA
.text:00B13BB8 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13BB8
.text:00B13BB8 loc_B13BB8:                             ; CODE XREF: .text:00B13BB2↑j
.text:00B13BB8                 mov     al, 1
.text:00B13BBA
.text:00B13BBA loc_B13BBA:                             ; CODE XREF: .text:00B13BB6↑j
.text:00B13BBA                 pop     ebp
.text:00B13BBB                 retn
.text:00B13BBB ; ───────────────────────────────────────────────────────────────────────────
.text:00B13BBC                 db 14h dup(0CCh)
.text:00B13BD0 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13BD0
.text:00B13BD0 loc_B13BD0:                             ; CODE XREF: .text:00B11280↑j
.text:00B13BD0                 push    ebp
.text:00B13BD1                 mov     ebp, esp
.text:00B13BD3                 call    sub_B11299
.text:00B13BD8                 movzx   eax, al
.text:00B13BDB                 test    eax, eax
.text:00B13BDD                 jnz     short loc_B13BE3
.text:00B13BDF                 xor     al, al
.text:00B13BE1                 jmp     short loc_B13BFA
.text:00B13BE3 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13BE3
.text:00B13BE3 loc_B13BE3:                             ; CODE XREF: .text:00B13BDD↑j
.text:00B13BE3                 call    sub_B112C6
.text:00B13BE8                 movzx   ecx, al
.text:00B13BEB                 test    ecx, ecx
.text:00B13BED                 jnz     short loc_B13BF8
.text:00B13BEF                 call    sub_B11136
.text:00B13BF4                 xor     al, al
.text:00B13BF6                 jmp     short loc_B13BFA
.text:00B13BF8 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13BF8
.text:00B13BF8 loc_B13BF8:                             ; CODE XREF: .text:00B13BED↑j
.text:00B13BF8                 mov     al, 1
.text:00B13BFA
.text:00B13BFA loc_B13BFA:                             ; CODE XREF: .text:00B13BE1↑j
.text:00B13BFA                                         ; .text:00B13BF6↑j
.text:00B13BFA                 pop     ebp
.text:00B13BFB                 retn
.text:00B13BFB ; ───────────────────────────────────────────────────────────────────────────
.text:00B13BFC                 db 14h dup(0CCh)
.text:00B13C10 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13C10
.text:00B13C10 loc_B13C10:                             ; CODE XREF: .text:00B112C1↑j
.text:00B13C10                 push    ebp
.text:00B13C11                 mov     ebp, esp
.text:00B13C13                 call    sub_B111D1
.text:00B13C18                 call    sub_B11136
.text:00B13C1D                 mov     al, 1
.text:00B13C1F                 pop     ebp
.text:00B13C20                 retn
.text:00B13C20 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13C21                 align 10h
.text:00B13C30
.text:00B13C30 loc_B13C30:                             ; CODE XREF: .text:00B1111D↑j
.text:00B13C30                 push    ebp
.text:00B13C31                 mov     ebp, esp
.text:00B13C33                 sub     esp, 8
.text:00B13C36                 call    sub_B111E5
```

```
.text:00B13C3B                 test    eax, eax
.text:00B13C3D                 jnz     short loc_B13C67
.text:00B13C3F                 cmp     dword ptr [ebp+0Ch], 1
.text:00B13C43                 jnz     short loc_B13C67
.text:00B13C45                 mov     eax, [ebp+14h]
.text:00B13C48                 mov     [ebp-8], eax
.text:00B13C4B                 mov     ecx, [ebp+10h]
.text:00B13C4E                 push    ecx
.text:00B13C4F                 push    0
.text:00B13C51                 mov     edx, [ebp+8]
.text:00B13C54                 push    edx
.text:00B13C55                 mov     eax, [ebp-8]
.text:00B13C58                 mov     [ebp-4], eax
.text:00B13C5B                 mov     ecx, [ebp-4]
.text:00B13C5E                 call    ds:___guard_check_icall_fptr
.text:00B13C64                 call    dword ptr [ebp-4]
.text:00B13C67
.text:00B13C67 loc_B13C67:                             ; CODE XREF: .text:00B13C3D↑j
.text:00B13C67                                         ; .text:00B13C43↑j
.text:00B13C67                 mov     ecx, [ebp+1Ch]
.text:00B13C6A                 push    ecx
.text:00B13C6B                 mov     edx, [ebp+18h]
.text:00B13C6E                 push    edx
.text:00B13C6F                 call    j__seh_filter_dll
.text:00B13C74                 add     esp, 8
.text:00B13C77                 mov     esp, ebp
.text:00B13C79                 pop     ebp
.text:00B13C7A                 retn
.text:00B13C7A ; ---------------------------------------------------------------------------
.text:00B13C7B                 db 15h dup(0CCh)
.text:00B13C90 ; ---------------------------------------------------------------------------
.text:00B13C90
.text:00B13C90 loc_B13C90:                             ; CODE XREF: .text:00B1112C↑j
.text:00B13C90                 push    ebp
.text:00B13C91                 mov     ebp, esp
.text:00B13C93                 call    sub_B111E5
.text:00B13C98                 test    eax, eax
.text:00B13C9A                 jz      short loc_B13CAB
.text:00B13C9C                 push    offset Table
.text:00B13CA1                 call    j__execute_onexit_table
.text:00B13CA6                 add     esp, 4
.text:00B13CA9                 jmp     short loc_B13CB9
.text:00B13CAB ; ---------------------------------------------------------------------------
.text:00B13CAB
.text:00B13CAB loc_B13CAB:                             ; CODE XREF: .text:00B13C9A↑j
.text:00B13CAB                 call    sub_B11235
.text:00B13CB0                 test    eax, eax
.text:00B13CB2                 jnz     short loc_B13CB9
.text:00B13CB4                 call    j__cexit
.text:00B13CB9
.text:00B13CB9 loc_B13CB9:                             ; CODE XREF: .text:00B13CA9↑j
.text:00B13CB9                                         ; .text:00B13CB2↑j
.text:00B13CB9                 pop     ebp
.text:00B13CBA                 retn
.text:00B13CBA ; ---------------------------------------------------------------------------
.text:00B13CBB                 db 15h dup(0CCh)
.text:00B13CD0 ; ---------------------------------------------------------------------------
.text:00B13CD0
.text:00B13CD0 loc_B13CD0:                             ; CODE XREF: .text:00B11078↑j
.text:00B13CD0                 push    ebp
.text:00B13CD1                 mov     ebp, esp
.text:00B13CD3                 push    0
.text:00B13CD5                 call    sub_B1141A
.text:00B13CDA                 add     esp, 4
.text:00B13CDD                 call    sub_B112A8
.text:00B13CE2                 pop     ebp
.text:00B13CE3                 retn
.text:00B13CE3 ; ---------------------------------------------------------------------------
.text:00B13CE4                 align 10h
.text:00B13CF0
.text:00B13CF0 ; =============== S U B R O U T I N E =======================================
.text:00B13CF0
.text:00B13CF0 ; Attributes: bp-based frame
.text:00B13CF0
.text:00B13CF0 sub_B13CF0      proc near               ; CODE XREF: sub_B1130C↑j
```

```
.text:00B13CF0
.text:00B13CF0 arg_0           = dword ptr  8
.text:00B13CF0
.text:00B13CF0                 push    ebp
.text:00B13CF1                 mov     ebp, esp
.text:00B13CF3                 cmp     [ebp+arg_0], 0
.text:00B13CF7                 jnz     short loc_B13D00
.text:00B13CF9                 mov     byte_B1C520, 1
.text:00B13D00
.text:00B13D00 loc_B13D00:                             ; CODE XREF: sub_B13CF0+7↑j
.text:00B13D00                 call    sub_B111EA
.text:00B13D05                 call    sub_B112DA
.text:00B13D0A                 movzx   eax, al
.text:00B13D0D                 test    eax, eax
.text:00B13D0F                 jnz     short loc_B13D15
.text:00B13D11                 xor     al, al
.text:00B13D13                 jmp     short loc_B13D31
.text:00B13D15 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13D15
.text:00B13D15 loc_B13D15:                             ; CODE XREF: sub_B13CF0+1F↑j
.text:00B13D15                 call    sub_B112DF
.text:00B13D1A                 movzx   ecx, al
.text:00B13D1D                 test    ecx, ecx
.text:00B13D1F                 jnz     short loc_B13D2F
.text:00B13D21                 push    0
.text:00B13D23                 call    sub_B1101E
.text:00B13D28                 add     esp, 4
.text:00B13D2B                 xor     al, al
.text:00B13D2D                 jmp     short loc_B13D31
.text:00B13D2F ; ───────────────────────────────────────────────────────────────────────────
.text:00B13D2F
.text:00B13D2F loc_B13D2F:                             ; CODE XREF: sub_B13CF0+2F↑j
.text:00B13D2F                 mov     al, 1
.text:00B13D31
.text:00B13D31 loc_B13D31:                             ; CODE XREF: sub_B13CF0+23↑j
.text:00B13D31                                         ; sub_B13CF0+3D↑j
.text:00B13D31                 pop     ebp
.text:00B13D32                 retn
.text:00B13D32 sub_B13CF0      endp
.text:00B13D32
.text:00B13D32 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13D33                 db 1Dh dup(0CCh)
.text:00B13D50
.text:00B13D50 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13D50
.text:00B13D50 ; Attributes: bp-based frame
.text:00B13D50
.text:00B13D50 sub_B13D50      proc near               ; CODE XREF: sub_B11168↑j
.text:00B13D50
.text:00B13D50 var_1C          = dword ptr -1Ch
.text:00B13D50 var_18          = dword ptr -18h
.text:00B13D50 var_14          = dword ptr -14h
.text:00B13D50 var_10          = dword ptr -10h
.text:00B13D50 var_C           = dword ptr -0Ch
.text:00B13D50 var_8           = dword ptr -8
.text:00B13D50 var_4           = dword ptr -4
.text:00B13D50 arg_0           = dword ptr  8
.text:00B13D50
.text:00B13D50                 push    ebp
.text:00B13D51                 mov     ebp, esp
.text:00B13D53                 sub     esp, 1Ch
.text:00B13D56                 movzx   eax, byte_B1C521
.text:00B13D5D                 test    eax, eax
.text:00B13D5F                 jz      short loc_B13D68
.text:00B13D61                 mov     al, 1
.text:00B13D63                 jmp     loc_B13E1E
.text:00B13D68 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13D68
.text:00B13D68 loc_B13D68:                             ; CODE XREF: sub_B13D50+F↑j
.text:00B13D68                 cmp     [ebp+arg_0], 0
.text:00B13D6C                 jz      short loc_B13D7B
.text:00B13D6E                 cmp     [ebp+arg_0], 1
.text:00B13D72                 jz      short loc_B13D7B
.text:00B13D74                 push    5
.text:00B13D76                 call    sub_B1121C
```

```
.text:00B13D7B
.text:00B13D7B loc_B13D7B:                             ; CODE XREF: sub_B13D50+1C↑j
.text:00B13D7B                                         ; sub_B13D50+22↑j
.text:00B13D7B                 call    sub_B111E5
.text:00B13D80                 test    eax, eax
.text:00B13D82                 jz      short loc_B13DB6
.text:00B13D84                 cmp     [ebp+arg_0], 0
.text:00B13D88                 jnz     short loc_B13DB6
.text:00B13D8A                 push    offset Table    ; Table
.text:00B13D8F                 call    j__initialize_onexit_table
.text:00B13D94                 add     esp, 4
.text:00B13D97                 test    eax, eax
.text:00B13D99                 jz      short loc_B13D9F
.text:00B13D9B                 xor     al, al
.text:00B13D9D                 jmp     short loc_B13E1E
.text:00B13D9F ; ───────────────────────────────────────────────────────────────────────────
.text:00B13D9F
.text:00B13D9F loc_B13D9F:                             ; CODE XREF: sub_B13D50+49↑j
.text:00B13D9F                 push    offset stru_B1C530 ; Table
.text:00B13DA4                 call    j__initialize_onexit_table
.text:00B13DA9                 add     esp, 4
.text:00B13DAC                 test    eax, eax
.text:00B13DAE                 jz      short loc_B13DB4
.text:00B13DB0                 xor     al, al
.text:00B13DB2                 jmp     short loc_B13E1E
.text:00B13DB4 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13DB4
.text:00B13DB4 loc_B13DB4:                             ; CODE XREF: sub_B13D50+5E↑j
.text:00B13DB4                 jmp     short loc_B13E15
.text:00B13DB6 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13DB6
.text:00B13DB6 loc_B13DB6:                             ; CODE XREF: sub_B13D50+32↑j
.text:00B13DB6                                         ; sub_B13D50+38↑j
.text:00B13DB6                 mov     [ebp+var_4], 0FFFFFFFFh
.text:00B13DBD                 mov     ecx, [ebp+var_4]
.text:00B13DC0                 mov     [ebp+var_10], ecx
.text:00B13DC3                 mov     edx, [ebp+var_4]
.text:00B13DC6                 mov     [ebp+var_C], edx
.text:00B13DC9                 mov     eax, [ebp+var_4]
.text:00B13DCC                 mov     [ebp+var_8], eax
.text:00B13DCF                 mov     ecx, [ebp+var_10]
.text:00B13DD2                 mov     Table._first, ecx
.text:00B13DD8                 mov     edx, [ebp+var_C]
.text:00B13DDB                 mov     Table._last, edx
.text:00B13DE1                 mov     eax, [ebp+var_8]
.text:00B13DE4                 mov     Table._end, eax
.text:00B13DE9                 mov     ecx, [ebp+var_4]
.text:00B13DEC                 mov     [ebp+var_1C], ecx
.text:00B13DEF                 mov     edx, [ebp+var_4]
.text:00B13DF2                 mov     [ebp+var_18], edx
.text:00B13DF5                 mov     eax, [ebp+var_4]
.text:00B13DF8                 mov     [ebp+var_14], eax
.text:00B13DFB                 mov     ecx, [ebp+var_1C]
.text:00B13DFE                 mov     stru_B1C530._first, ecx
.text:00B13E04                 mov     edx, [ebp+var_18]
.text:00B13E07                 mov     stru_B1C530._last, edx
.text:00B13E0D                 mov     eax, [ebp+var_14]
.text:00B13E10                 mov     stru_B1C530._end, eax
.text:00B13E15
.text:00B13E15 loc_B13E15:                             ; CODE XREF: sub_B13D50:loc_B13DB4↑j
.text:00B13E15                 mov     byte_B1C521, 1
.text:00B13E1C                 mov     al, 1
.text:00B13E1E
.text:00B13E1E loc_B13E1E:                             ; CODE XREF: sub_B13D50+13↑j
.text:00B13E1E                                         ; sub_B13D50+4D↑j ...
.text:00B13E1E                 mov     esp, ebp
.text:00B13E20                 pop     ebp
.text:00B13E21                 retn
.text:00B13E21 sub_B13D50      endp ; sp-analysis failed
.text:00B13E21
.text:00B13E21 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13E22                 db 3Eh dup(0CCh)
.text:00B13E60
.text:00B13E60 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13E60
```

```
.text:00B13E60 ; Attributes: bp-based frame
.text:00B13E60
.text:00B13E60 sub_B13E60      proc near               ; CODE XREF: sub_B11154↑j
.text:00B13E60
.text:00B13E60 var_38          = dword ptr -38h
.text:00B13E60 var_34          = dword ptr -34h
.text:00B13E60 var_30          = dword ptr -30h
.text:00B13E60 var_2C          = dword ptr -2Ch
.text:00B13E60 var_28          = dword ptr -28h
.text:00B13E60 var_24          = dword ptr -24h
.text:00B13E60 var_1D          = byte ptr -1Dh
.text:00B13E60 var_1C          = byte ptr -1Ch
.text:00B13E60 var_1B          = byte ptr -1Bh
.text:00B13E60 var_1A          = byte ptr -1Ah
.text:00B13E60 var_19          = byte ptr -19h
.text:00B13E60 ms_exc          = CPPEH_RECORD ptr -18h
.text:00B13E60 arg_0           = dword ptr  8
.text:00B13E60
.text:00B13E60                 push    ebp
.text:00B13E61                 mov     ebp, esp
.text:00B13E63                 push    0FFFFFFFEh
.text:00B13E65                 push    offset stru_B1B2F8
.text:00B13E6A                 push    offset SEH_4128D0
.text:00B13E6F                 mov     eax, large fs:0
.text:00B13E75                 push    eax
.text:00B13E76                 add     esp, 0FFFFFFD8h
.text:00B13E79                 push    ebx
.text:00B13E7A                 push    esi
.text:00B13E7B                 push    edi
.text:00B13E7C                 mov     eax, ___security_cookie
.text:00B13E81                 xor     [ebp+ms_exc.registration.ScopeTable], eax
.text:00B13E84                 xor     eax, ebp
.text:00B13E86                 push    eax
.text:00B13E87                 lea     eax, [ebp+ms_exc.registration]
.text:00B13E8A                 mov     large fs:0, eax
.text:00B13E90                 mov     [ebp+ms_exc.old_esp], esp
.text:00B13E93                 mov     eax, [ebp+arg_0]
.text:00B13E96                 mov     [ebp+var_30], eax
.text:00B13E99                 mov     [ebp+var_24], 0B00000h
.text:00B13EA0                 mov     [ebp+ms_exc.registration.TryLevel], 0
.text:00B13EA7                 mov     ecx, [ebp+var_24]
.text:00B13EAA                 push    ecx
.text:00B13EAB                 call    sub_B13A60
.text:00B13EB0                 add     esp, 4
.text:00B13EB3                 movzx   edx, al
.text:00B13EB6                 test    edx, edx
.text:00B13EB8                 jnz     short loc_B13ECD
.text:00B13EBA                 mov     [ebp+var_19], 0
.text:00B13EBE                 mov     [ebp+ms_exc.registration.TryLevel], 0FFFFFFFEh
.text:00B13EC5                 mov     al, [ebp+var_19]
.text:00B13EC8                 jmp     loc_B13F77
.text:00B13ECD ; ───────────────────────────────────────────────────────────────────────────
.text:00B13ECD
.text:00B13ECD loc_B13ECD:                             ; CODE XREF: sub_B13E60+58↑j
.text:00B13ECD                 mov     eax, [ebp+var_30]
.text:00B13ED0                 sub     eax, [ebp+var_24]
.text:00B13ED3                 mov     [ebp+var_34], eax
.text:00B13ED6                 mov     ecx, [ebp+var_34]
.text:00B13ED9                 push    ecx
.text:00B13EDA                 mov     edx, [ebp+var_24]
.text:00B13EDD                 push    edx
.text:00B13EDE                 call    sub_B139B0
.text:00B13EE3                 add     esp, 8
.text:00B13EE6                 mov     [ebp+var_28], eax
.text:00B13EE9                 cmp     [ebp+var_28], 0
.text:00B13EED                 jnz     short loc_B13EFF
.text:00B13EEF                 mov     [ebp+var_1A], 0
.text:00B13EF3                 mov     [ebp+ms_exc.registration.TryLevel], 0FFFFFFFEh
.text:00B13EFA                 mov     al, [ebp+var_1A]
.text:00B13EFD                 jmp     short loc_B13F77
.text:00B13EFF ; ───────────────────────────────────────────────────────────────────────────
.text:00B13EFF
.text:00B13EFF loc_B13EFF:                             ; CODE XREF: sub_B13E60+8D↑j
.text:00B13EFF                 mov     eax, [ebp+var_28]
.text:00B13F02                 mov     ecx, [eax+24h]
```

```
.text:00B13F05                and     ecx, 80000000h
.text:00B13F0B                jz      short loc_B13F1D
.text:00B13F0D                mov     [ebp+var_1B], 0
.text:00B13F11                mov     [ebp+ms_exc.registration.TryLevel], 0FFFFFFFEh
.text:00B13F18                mov     al, [ebp+var_1B]
.text:00B13F1B                jmp     short loc_B13F77
.text:00B13F1D ; ───────────────────────────────────────────────────────────────────────────
.text:00B13F1D
.text:00B13F1D loc_B13F1D:                            ; CODE XREF: sub_B13E60+AB↑j
.text:00B13F1D                mov     [ebp+var_1C], 1
.text:00B13F21                mov     [ebp+ms_exc.registration.TryLevel], 0FFFFFFFEh
.text:00B13F28                mov     al, [ebp+var_1C]
.text:00B13F2B                jmp     short loc_B13F77
.text:00B13F2D ; ───────────────────────────────────────────────────────────────────────────
.text:00B13F2D                mov     [ebp+ms_exc.registration.TryLevel], 0FFFFFFFEh
.text:00B13F34                jmp     short loc_B13F77
.text:00B13F36 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13F36
.text:00B13F36 loc_B13F36:                            ; DATA XREF: .rdata:stru_B1B2F8↓o
.text:00B13F36                mov     edx, [ebp+ms_exc.exc_ptr]
.text:00B13F39                mov     eax, [edx]
.text:00B13F3B                mov     ecx, [eax]
.text:00B13F3D                mov     [ebp+var_38], ecx
.text:00B13F40                cmp     [ebp+var_38], 0C0000005h
.text:00B13F47                jnz     short loc_B13F52
.text:00B13F49                mov     [ebp+var_2C], 1
.text:00B13F50                jmp     short loc_B13F59
.text:00B13F52 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13F52
.text:00B13F52 loc_B13F52:                            ; CODE XREF: sub_B13E60+E7↑j
.text:00B13F52                mov     [ebp+var_2C], 0
.text:00B13F59
.text:00B13F59 loc_B13F59:                            ; CODE XREF: sub_B13E60+F0↑j
.text:00B13F59                mov     eax, [ebp+var_2C]
.text:00B13F5C                retn
.text:00B13F5D ; ───────────────────────────────────────────────────────────────────────────
.text:00B13F5D
.text:00B13F5D loc_B13F5D:                            ; DATA XREF: .rdata:stru_B1B2F8↓o
.text:00B13F5D                mov     esp, [ebp+ms_exc.old_esp]
.text:00B13F60                mov     [ebp+var_1D], 0
.text:00B13F64                mov     [ebp+ms_exc.registration.TryLevel], 0FFFFFFFEh
.text:00B13F6B                mov     al, [ebp+var_1D]
.text:00B13F6E                jmp     short loc_B13F77
.text:00B13F70 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13F70                mov     [ebp+ms_exc.registration.TryLevel], 0FFFFFFFEh
.text:00B13F77
.text:00B13F77 loc_B13F77:                            ; CODE XREF: sub_B13E60+68↑j
.text:00B13F77                                        ; sub_B13E60+9D↑j ...
.text:00B13F77                mov     ecx, [ebp+ms_exc.registration.Next]
.text:00B13F7A                mov     large fs:0, ecx
.text:00B13F81                pop     ecx
.text:00B13F82                pop     edi
.text:00B13F83                pop     esi
.text:00B13F84                pop     ebx
.text:00B13F85                mov     esp, ebp
.text:00B13F87                pop     ebp
.text:00B13F88                retn
.text:00B13F88 sub_B13E60      endp
.text:00B13F88
.text:00B13F88 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13F89                db 57h dup(0CCh)
.text:00B13FE0
.text:00B13FE0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B13FE0
.text:00B13FE0 ; Attributes: bp-based frame
.text:00B13FE0
.text:00B13FE0 sub_B13FE0      proc near               ; CODE XREF: sub_B111AE↑j
.text:00B13FE0
.text:00B13FE0 arg_0           = byte ptr  8
.text:00B13FE0
.text:00B13FE0                push    ebp
.text:00B13FE1                mov     ebp, esp
.text:00B13FE3                call    sub_B111E5
.text:00B13FE8                test    eax, eax
.text:00B13FEA                jnz     short loc_B13FEE
```

```
.text:00B13FEC                 jmp     short loc_B14001
.text:00B13FEE ; ───────────────────────────────────────────────────────────────────────────
.text:00B13FEE
.text:00B13FEE loc_B13FEE:                             ; CODE XREF: sub_B13FE0+A↑j
.text:00B13FEE                 movzx   eax, [ebp+arg_0]
.text:00B13FF2                 test    eax, eax
.text:00B13FF4                 jz      short loc_B13FF8
.text:00B13FF6                 jmp     short loc_B14001
.text:00B13FF8 ; ───────────────────────────────────────────────────────────────────────────
.text:00B13FF8
.text:00B13FF8 loc_B13FF8:                             ; CODE XREF: sub_B13FE0+14↑j
.text:00B13FF8                 xor     ecx, ecx
.text:00B13FFA                 mov     edx, offset unk_B1C51C
.text:00B13FFF                 xchg    ecx, [edx]
.text:00B14001
.text:00B14001 loc_B14001:                             ; CODE XREF: sub_B13FE0+C↑j
.text:00B14001                                         ; sub_B13FE0+16↑j
.text:00B14001                 pop     ebp
.text:00B14002                 retn
.text:00B14002 sub_B13FE0      endp
.text:00B14002
.text:00B14002 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14003                 align 10h
.text:00B14010
.text:00B14010 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14010
.text:00B14010 ; Attributes: bp-based frame
.text:00B14010
.text:00B14010 sub_B14010      proc near               ; CODE XREF: sub_B111C2↑j
.text:00B14010
.text:00B14010 arg_0           = byte ptr  8
.text:00B14010 arg_4           = byte ptr  0Ch
.text:00B14010
.text:00B14010                 push    ebp
.text:00B14011                 mov     ebp, esp
.text:00B14013                 movzx   eax, byte_B1C520
.text:00B1401A                 test    eax, eax
.text:00B1401C                 jz      short loc_B1402A
.text:00B1401E                 movzx   ecx, [ebp+arg_4]
.text:00B14022                 test    ecx, ecx
.text:00B14024                 jz      short loc_B1402A
.text:00B14026                 mov     al, 1
.text:00B14028                 jmp     short loc_B14046
.text:00B1402A ; ───────────────────────────────────────────────────────────────────────────
.text:00B1402A
.text:00B1402A loc_B1402A:                             ; CODE XREF: sub_B14010+C↑j
.text:00B1402A                                         ; sub_B14010+14↑j
.text:00B1402A                 movzx   edx, [ebp+arg_0]
.text:00B1402E                 push    edx
.text:00B1402F                 call    sub_B1115E
.text:00B14034                 add     esp, 4
.text:00B14037                 movzx   eax, [ebp+arg_0]
.text:00B1403B                 push    eax
.text:00B1403C                 call    sub_B1101E
.text:00B14041                 add     esp, 4
.text:00B14044                 mov     al, 1
.text:00B14046
.text:00B14046 loc_B14046:                             ; CODE XREF: sub_B14010+18↑j
.text:00B14046                 pop     ebp
.text:00B14047                 retn
.text:00B14047 sub_B14010      endp
.text:00B14047
.text:00B14047 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14048                 align 20h
.text:00B14060
.text:00B14060 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14060
.text:00B14060 ; Attributes: bp-based frame
.text:00B14060
.text:00B14060 ; int __cdecl sub_B14060(_PVFV Function)
.text:00B14060 sub_B14060      proc near               ; CODE XREF: sub_B11262↑j
.text:00B14060
.text:00B14060 var_C           = dword ptr -0Ch
.text:00B14060 var_8           = dword ptr -8
.text:00B14060 var_4           = dword ptr -4
```

```
.text:00B14060 Function        = dword ptr  8
.text:00B14060
.text:00B14060                 push    ebp
.text:00B14061                 mov     ebp, esp
.text:00B14063                 sub     esp, 0Ch
.text:00B14066                 mov     eax, Table._first
.text:00B1406B                 mov     [ebp+var_C], eax
.text:00B1406E                 cmp     [ebp+var_C], 0FFFFFFFFh
.text:00B14072                 jnz     short loc_B1409A
.text:00B14074                 mov     ecx, [ebp+Function]
.text:00B14077                 push    ecx             ; Function
.text:00B14078                 call    j__crt_atexit
.text:00B1407D                 add     esp, 4
.text:00B14080                 test    eax, eax
.text:00B14082                 jnz     short loc_B1408C
.text:00B14084                 mov     edx, [ebp+Function]
.text:00B14087                 mov     [ebp+var_4], edx
.text:00B1408A                 jmp     short loc_B14093
.text:00B1408C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1408C
.text:00B1408C loc_B1408C:                             ; CODE XREF: sub_B14060+22↑j
.text:00B1408C                 mov     [ebp+var_4], 0
.text:00B14093
.text:00B14093 loc_B14093:                             ; CODE XREF: sub_B14060+2A↑j
.text:00B14093                 mov     eax, [ebp+var_4]
.text:00B14096                 jmp     short loc_B140C1
.text:00B14098 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14098                 jmp     short loc_B140C1
.text:00B1409A ; ───────────────────────────────────────────────────────────────────────────
.text:00B1409A
.text:00B1409A loc_B1409A:                             ; CODE XREF: sub_B14060+12↑j
.text:00B1409A                 mov     eax, [ebp+Function]
.text:00B1409D                 push    eax             ; Function
.text:00B1409E                 push    offset Table    ; Table
.text:00B140A3                 call    j__register_onexit_function
.text:00B140A8                 add     esp, 8
.text:00B140AB                 test    eax, eax
.text:00B140AD                 jnz     short loc_B140B7
.text:00B140AF                 mov     ecx, [ebp+Function]
.text:00B140B2                 mov     [ebp+var_8], ecx
.text:00B140B5                 jmp     short loc_B140BE
.text:00B140B7 ; ───────────────────────────────────────────────────────────────────────────
.text:00B140B7
.text:00B140B7 loc_B140B7:                             ; CODE XREF: sub_B14060+4D↑j
.text:00B140B7                 mov     [ebp+var_8], 0
.text:00B140BE
.text:00B140BE loc_B140BE:                             ; CODE XREF: sub_B14060+55↑j
.text:00B140BE                 mov     eax, [ebp+var_8]
.text:00B140C1
.text:00B140C1 loc_B140C1:                             ; CODE XREF: sub_B14060+36↑j
.text:00B140C1                                         ; sub_B14060+38↑j
.text:00B140C1                 mov     esp, ebp
.text:00B140C3                 pop     ebp
.text:00B140C4                 retn
.text:00B140C4 sub_B14060      endp
.text:00B140C4
.text:00B140C4 ; ───────────────────────────────────────────────────────────────────────────
.text:00B140C5                 align 20h
.text:00B140E0
.text:00B140E0 loc_B140E0:                             ; CODE XREF: .text:00B1126C↑j
.text:00B140E0                 push    ebp
.text:00B140E1                 mov     ebp, esp
.text:00B140E3                 sub     esp, 8
.text:00B140E6                 mov     eax, stru_B1C530._first
.text:00B140EB                 mov     [ebp-4], eax
.text:00B140EE                 cmp     dword ptr [ebp-4], 0FFFFFFFFh
.text:00B140F2                 jnz     short loc_B14104
.text:00B140F4                 mov     ecx, [ebp+8]
.text:00B140F7                 push    ecx
.text:00B140F8                 call    j__crt_at_quick_exit
.text:00B140FD                 add     esp, 4
.text:00B14100                 jmp     short loc_B1411B
.text:00B14102 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14102                 jmp     short loc_B1411B
.text:00B14104 ; ───────────────────────────────────────────────────────────────────────────
```

```
.text:00B14104
.text:00B14104 loc_B14104:                             ; CODE XREF: .text:00B140F2↑j
.text:00B14104                 mov     edx, [ebp+8]
.text:00B14107                 mov     [ebp-8], edx
.text:00B1410A                 mov     eax, [ebp-8]
.text:00B1410D                 push    eax
.text:00B1410E                 push    offset stru_B1C530
.text:00B14113                 call    j__register_onexit_function
.text:00B14118                 add     esp, 8
.text:00B1411B
.text:00B1411B loc_B1411B:                             ; CODE XREF: .text:00B14100↑j
.text:00B1411B                                         ; .text:00B14102↑j
.text:00B1411B                 mov     esp, ebp
.text:00B1411D                 pop     ebp
.text:00B1411E                 retn
.text:00B1411E ; ───────────────────────────────────────────────────────────────────────────
.text:00B1411F                 db 11h dup(0CCh)
.text:00B14130
.text:00B14130 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14130
.text:00B14130 ; Attributes: bp-based frame
.text:00B14130
.text:00B14130 ; int __cdecl sub_B14130(_PVFV Function)
.text:00B14130 sub_B14130      proc near               ; CODE XREF: sub_B1123A↑j
.text:00B14130
.text:00B14130 var_4           = dword ptr -4
.text:00B14130 Function        = dword ptr  8
.text:00B14130
.text:00B14130                 push    ebp
.text:00B14131                 mov     ebp, esp
.text:00B14133                 push    ecx
.text:00B14134                 mov     eax, [ebp+Function]
.text:00B14137                 push    eax             ; Function
.text:00B14138                 call    sub_B11262
.text:00B1413D                 add     esp, 4
.text:00B14140                 test    eax, eax
.text:00B14142                 jz      short loc_B1414D
.text:00B14144                 mov     [ebp+var_4], 0
.text:00B1414B                 jmp     short loc_B14154
.text:00B1414D ; ───────────────────────────────────────────────────────────────────────────
.text:00B1414D
.text:00B1414D loc_B1414D:                             ; CODE XREF: sub_B14130+12↑j
.text:00B1414D                 mov     [ebp+var_4], 0FFFFFFFFh
.text:00B14154
.text:00B14154 loc_B14154:                             ; CODE XREF: sub_B14130+1B↑j
.text:00B14154                 mov     eax, [ebp+var_4]
.text:00B14157                 mov     esp, ebp
.text:00B14159                 pop     ebp
.text:00B1415A                 retn
.text:00B1415A sub_B14130      endp
.text:00B1415A
.text:00B1415A ; ───────────────────────────────────────────────────────────────────────────
.text:00B1415B                 db 15h dup(0CCh)
.text:00B14170
.text:00B14170 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14170
.text:00B14170 ; Attributes: bp-based frame
.text:00B14170
.text:00B14170 ; int __cdecl sub_B14170(_DWORD)
.text:00B14170 sub_B14170      proc near               ; CODE XREF: sub_B14200:loc_B1422C↓p
.text:00B14170
.text:00B14170 PerformanceCount= LARGE_INTEGER ptr -14h
.text:00B14170 SystemTimeAsFileTime= _FILETIME ptr -0Ch
.text:00B14170 var_4           = dword ptr -4
.text:00B14170
.text:00B14170                 push    ebp
.text:00B14171                 mov     ebp, esp
.text:00B14173                 sub     esp, 14h
.text:00B14176                 xor     eax, eax
.text:00B14178                 mov     [ebp+SystemTimeAsFileTime.dwLowDateTime], eax
.text:00B1417B                 mov     [ebp+SystemTimeAsFileTime.dwHighDateTime], eax
.text:00B1417E                 lea     ecx, [ebp+SystemTimeAsFileTime]
.text:00B14181                 push    ecx             ; lpSystemTimeAsFileTime
.text:00B14182                 call    ds:GetSystemTimeAsFileTime
.text:00B14188                 mov     edx, [ebp+SystemTimeAsFileTime.dwLowDateTime]
```

```
.text:00B1418B                 mov     [ebp+var_4], edx
.text:00B1418E                 mov     eax, [ebp+var_4]
.text:00B14191                 xor     eax, [ebp+SystemTimeAsFileTime.dwHighDateTime]
.text:00B14194                 mov     [ebp+var_4], eax
.text:00B14197                 call    ds:GetCurrentThreadId
.text:00B1419D                 xor     eax, [ebp+var_4]
.text:00B141A0                 mov     [ebp+var_4], eax
.text:00B141A3                 call    ds:GetCurrentProcessId
.text:00B141A9                 xor     eax, [ebp+var_4]
.text:00B141AC                 mov     [ebp+var_4], eax
.text:00B141AF                 lea     ecx, [ebp+PerformanceCount]
.text:00B141B2                 push    ecx             ; lpPerformanceCount
.text:00B141B3                 call    ds:QueryPerformanceCounter
.text:00B141B9                 mov     edx, [ebp+var_4]
.text:00B141BC                 xor     edx, dword ptr [ebp+PerformanceCount]
.text:00B141BF                 mov     [ebp+var_4], edx
.text:00B141C2                 mov     eax, [ebp+var_4]
.text:00B141C5                 xor     eax, dword ptr [ebp+PerformanceCount+4]
.text:00B141C8                 mov     [ebp+var_4], eax
.text:00B141CB                 mov     ecx, [ebp+var_4]
.text:00B141CE                 lea     edx, [ebp+var_4]
.text:00B141D1                 xor     ecx, edx
.text:00B141D3                 mov     [ebp+var_4], ecx
.text:00B141D6                 mov     eax, [ebp+var_4]
.text:00B141D9                 mov     esp, ebp
.text:00B141DB                 pop     ebp
.text:00B141DC                 retn
.text:00B141DC sub_B14170      endp
.text:00B141DC
.text:00B141DC ; ───────────────────────────────────────────────────────────────────────────
.text:00B141DD                 align 40h
.text:00B14200
.text:00B14200 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14200
.text:00B14200 ; Attributes: bp-based frame
.text:00B14200
.text:00B14200 ; uintptr_t __thiscall sub_B14200(void *this)
.text:00B14200 sub_B14200      proc near               ; CODE XREF: sub_B11415↑j
.text:00B14200
.text:00B14200 var_4           = dword ptr -4
.text:00B14200
.text:00B14200                 push    ebp
.text:00B14201                 mov     ebp, esp
.text:00B14203                 push    ecx
.text:00B14204                 cmp     ___security_cookie, 0BB40E64Eh
.text:00B1420E                 jz      short loc_B1422C
.text:00B14210                 mov     eax, ___security_cookie
.text:00B14215                 and     eax, 0FFFF0000h
.text:00B1421A                 jz      short loc_B1422C
.text:00B1421C                 mov     ecx, ___security_cookie
.text:00B14222                 not     ecx
.text:00B14224                 mov     dword_B1C000, ecx
.text:00B1422A                 jmp     short loc_B14276
.text:00B1422C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1422C
.text:00B1422C loc_B1422C:                             ; CODE XREF: sub_B14200+E↑j
.text:00B1422C                                         ; sub_B14200+1A↑j
.text:00B1422C                 call    sub_B14170
.text:00B14231                 mov     [ebp+var_4], eax
.text:00B14234                 cmp     [ebp+var_4], 0BB40E64Eh
.text:00B1423B                 jnz     short loc_B14246
.text:00B1423D                 mov     [ebp+var_4], 0BB40E64Fh
.text:00B14244                 jmp     short loc_B14262
.text:00B14246 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14246
.text:00B14246 loc_B14246:                             ; CODE XREF: sub_B14200+3B↑j
.text:00B14246                 mov     edx, [ebp+var_4]
.text:00B14249                 and     edx, 0FFFF0000h
.text:00B1424F                 jnz     short loc_B14262
.text:00B14251                 mov     eax, [ebp+var_4]
.text:00B14254                 or      eax, 4711h
.text:00B14259                 shl     eax, 10h
.text:00B1425C                 or      eax, [ebp+var_4]
.text:00B1425F                 mov     [ebp+var_4], eax
.text:00B14262
```

```
.text:00B14262 loc_B14262:                             ; CODE XREF: sub_B14200+44↑j
.text:00B14262                                         ; sub_B14200+4F↑j
.text:00B14262                 mov     ecx, [ebp+var_4]
.text:00B14265                 mov     ___security_cookie, ecx
.text:00B1426B                 mov     edx, [ebp+var_4]
.text:00B1426E                 not     edx
.text:00B14270                 mov     dword_B1C000, edx
.text:00B14276
.text:00B14276 loc_B14276:                             ; CODE XREF: sub_B14200+2A↑j
.text:00B14276                 mov     esp, ebp
.text:00B14278                 pop     ebp
.text:00B14279                 retn
.text:00B14279 sub_B14200      endp
.text:00B14279
.text:00B14279 ; ---------------------------------------------------------------------------
.text:00B1427A                 db 26h dup(0CCh)
.text:00B142A0
.text:00B142A0 ; =============== S U B R O U T I N E =======================================
.text:00B142A0
.text:00B142A0 ; Attributes: bp-based frame
.text:00B142A0
.text:00B142A0 sub_B142A0      proc near               ; CODE XREF: UserMathErrorFunction↑j
.text:00B142A0                 push    ebp
.text:00B142A1                 mov     ebp, esp
.text:00B142A3                 xor     eax, eax
.text:00B142A5                 pop     ebp
.text:00B142A6                 retn
.text:00B142A6 sub_B142A0      endp
.text:00B142A6
.text:00B142A6 ; ---------------------------------------------------------------------------
.text:00B142A7                 align 10h
.text:00B142B0
.text:00B142B0 ; =============== S U B R O U T I N E =======================================
.text:00B142B0
.text:00B142B0 ; Attributes: bp-based frame
.text:00B142B0
.text:00B142B0 sub_B142B0      proc near               ; CODE XREF: sub_B112EE↑j
.text:00B142B0                 push    ebp
.text:00B142B1                 mov     ebp, esp
.text:00B142B3                 mov     eax, 1
.text:00B142B8                 pop     ebp
.text:00B142B9                 retn
.text:00B142B9 sub_B142B0      endp
.text:00B142B9
.text:00B142B9 ; ---------------------------------------------------------------------------
.text:00B142BA                 align 10h
.text:00B142C0
.text:00B142C0 ; =============== S U B R O U T I N E =======================================
.text:00B142C0
.text:00B142C0 ; Attributes: bp-based frame
.text:00B142C0
.text:00B142C0 sub_B142C0      proc near               ; CODE XREF: sub_B1114F↑j
.text:00B142C0                 push    ebp
.text:00B142C1                 mov     ebp, esp
.text:00B142C3                 xor     eax, eax
.text:00B142C5                 pop     ebp
.text:00B142C6                 retn
.text:00B142C6 sub_B142C0      endp
.text:00B142C6
.text:00B142C6 ; ---------------------------------------------------------------------------
.text:00B142C7                 align 10h
.text:00B142D0
.text:00B142D0 ; =============== S U B R O U T I N E =======================================
.text:00B142D0
.text:00B142D0 ; Attributes: bp-based frame
.text:00B142D0
.text:00B142D0 sub_B142D0      proc near               ; CODE XREF: sub_B110FA↑j
.text:00B142D0                 push    ebp
.text:00B142D1                 mov     ebp, esp
.text:00B142D3                 mov     eax, 4000h
.text:00B142D8                 pop     ebp
.text:00B142D9                 retn
.text:00B142D9 sub_B142D0      endp
.text:00B142D9
.text:00B142D9 ; ---------------------------------------------------------------------------
```

```
.text:00B142DA                 align 10h
.text:00B142E0
.text:00B142E0 ; =============== S U B R O U T I N E =======================================
.text:00B142E0
.text:00B142E0 ; Attributes: bp-based frame
.text:00B142E0
.text:00B142E0 sub_B142E0      proc near               ; CODE XREF: sub_B1117C↑j
.text:00B142E0                 push    ebp
.text:00B142E1                 mov     ebp, esp
.text:00B142E3                 xor     eax, eax
.text:00B142E5                 pop     ebp
.text:00B142E6                 retn
.text:00B142E6 sub_B142E0      endp
.text:00B142E6
.text:00B142E6 ; ---------------------------------------------------------------------------
.text:00B142E7                 align 10h
.text:00B142F0
.text:00B142F0 ; =============== S U B R O U T I N E =======================================
.text:00B142F0
.text:00B142F0 ; Attributes: bp-based frame
.text:00B142F0
.text:00B142F0 sub_B142F0      proc near               ; CODE XREF: sub_B112D0↑j
.text:00B142F0                 push    ebp
.text:00B142F1                 mov     ebp, esp
.text:00B142F3                 xor     eax, eax
.text:00B142F5                 pop     ebp
.text:00B142F6                 retn
.text:00B142F6 sub_B142F0      endp
.text:00B142F6
.text:00B142F6 ; ---------------------------------------------------------------------------
.text:00B142F7                 align 10h
.text:00B14300
.text:00B14300 ; =============== S U B R O U T I N E =======================================
.text:00B14300
.text:00B14300 ; Attributes: bp-based frame
.text:00B14300
.text:00B14300 sub_B14300      proc near               ; CODE XREF: sub_B110AA↑j
.text:00B14300                 push    ebp
.text:00B14301                 mov     ebp, esp
.text:00B14303                 push    offset ListHead ; ListHead
.text:00B14308                 call    ds:InitializeSListHead
.text:00B1430E                 pop     ebp
.text:00B1430F                 retn
.text:00B1430F sub_B14300      endp
.text:00B1430F
.text:00B1430F ; ---------------------------------------------------------------------------
.text:00B14310                 align 20h
.text:00B14320
.text:00B14320 loc_B14320:                             ; CODE XREF: .text:00B112BC↑j
.text:00B14320                 push    ebp
.text:00B14321                 mov     ebp, esp
.text:00B14323                 push    offset ListHead
.text:00B14328                 call    j___std_type_info_destroy_list
.text:00B1432D                 add     esp, 4
.text:00B14330                 pop     ebp
.text:00B14331                 retn
.text:00B14331 ; ---------------------------------------------------------------------------
.text:00B14332                 align 10h
.text:00B14340
.text:00B14340 ; =============== S U B R O U T I N E =======================================
.text:00B14340
.text:00B14340 ; Attributes: bp-based frame
.text:00B14340
.text:00B14340 sub_B14340      proc near               ; CODE XREF: sub_B1134D↑j
.text:00B14340                 push    ebp
.text:00B14341                 mov     ebp, esp
.text:00B14343                 mov     al, 1
.text:00B14345                 pop     ebp
.text:00B14346                 retn
.text:00B14346 sub_B14340      endp
.text:00B14346
.text:00B14346 ; ---------------------------------------------------------------------------
.text:00B14347                 align 10h
.text:00B14350
.text:00B14350 ; =============== S U B R O U T I N E =======================================
.text:00B14350
.text:00B14350 ; Attributes: bp-based frame
.text:00B14350
.text:00B14350 sub_B14350      proc near               ; CODE XREF: sub_B11389↑j
.text:00B14350                 push    ebp
.text:00B14351                 mov     ebp, esp
.text:00B14353                 push    30000h          ; Mask
.text:00B14358                 push    10000h          ; NewValue
.text:00B1435D                 push    0               ; CurrentState
.text:00B1435F                 call    j__controlfp_s
.text:00B14364                 add     esp, 0Ch
.text:00B14367                 test    eax, eax
.text:00B14369                 jz      short loc_B14372
.text:00B1436B                 push    7
.text:00B1436D                 call    sub_B1121C
.text:00B14372
.text:00B14372 loc_B14372:                             ; CODE XREF: sub_B14350+19↑j
.text:00B14372                 pop     ebp
.text:00B14373                 retn
.text:00B14373 sub_B14350      endp ; sp-analysis failed
.text:00B14373
.text:00B14373 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14374                 align 10h
.text:00B14380
.text:00B14380 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14380
.text:00B14380 ; Attributes: bp-based frame
.text:00B14380
.text:00B14380 sub_B14380      proc near               ; CODE XREF: sub_B113FC↑j
.text:00B14380                 push    ebp
.text:00B14381                 mov     ebp, esp
.text:00B14383                 pop     ebp
.text:00B14384                 retn
.text:00B14384 sub_B14380      endp
.text:00B14384
.text:00B14384 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14385                 align 10h
.text:00B14390
.text:00B14390 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14390
.text:00B14390 ; Attributes: bp-based frame
.text:00B14390
.text:00B14390 sub_B14390      proc near               ; CODE XREF: sub_B1103C↑j
.text:00B14390                 push    ebp
.text:00B14391                 mov     ebp, esp
.text:00B14393                 pop     ebp
.text:00B14394                 retn
.text:00B14394 sub_B14390      endp
.text:00B14394
.text:00B14394 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14395                 align 10h
.text:00B143A0
.text:00B143A0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B143A0
.text:00B143A0 ; Attributes: bp-based frame
.text:00B143A0
.text:00B143A0 sub_B143A0      proc near               ; CODE XREF: sub_B113BB↑j
.text:00B143A0                 push    ebp
.text:00B143A1                 mov     ebp, esp
.text:00B143A3                 mov     eax, offset unk_B1C558
.text:00B143A8                 pop     ebp
.text:00B143A9                 retn
.text:00B143A9 sub_B143A0      endp
.text:00B143A9
.text:00B143A9 ; ───────────────────────────────────────────────────────────────────────────
.text:00B143AA                 align 10h
.text:00B143B0
.text:00B143B0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B143B0
.text:00B143B0 ; Attributes: bp-based frame
.text:00B143B0
.text:00B143B0 sub_B143B0      proc near               ; CODE XREF: sub_B111A4↑j
.text:00B143B0
.text:00B143B0 var_8           = dword ptr -8
.text:00B143B0 var_4           = dword ptr -4
```

```
.text:00B143B0
.text:00B143B0                 push    ebp
.text:00B143B1                 mov     ebp, esp
.text:00B143B3                 sub     esp, 8
.text:00B143B6                 call    sub_B1135C
.text:00B143BB                 mov     [ebp+var_4], eax
.text:00B143BE                 mov     eax, [ebp+var_4]
.text:00B143C1                 mov     ecx, [eax]
.text:00B143C3                 or      ecx, 24h
.text:00B143C6                 mov     edx, [eax+4]
.text:00B143C9                 mov     eax, [ebp+var_4]
.text:00B143CC                 mov     [eax], ecx
.text:00B143CE                 mov     [eax+4], edx
.text:00B143D1                 call    sub_B113BB
.text:00B143D6                 mov     [ebp+var_8], eax
.text:00B143D9                 mov     ecx, [ebp+var_8]
.text:00B143DC                 mov     edx, [ecx]
.text:00B143DE                 or      edx, 2
.text:00B143E1                 mov     eax, [ecx+4]
.text:00B143E4                 mov     ecx, [ebp+var_8]
.text:00B143E7                 mov     [ecx], edx
.text:00B143E9                 mov     [ecx+4], eax
.text:00B143EC                 mov     esp, ebp
.text:00B143EE                 pop     ebp
.text:00B143EF                 retn
.text:00B143EF sub_B143B0      endp
.text:00B143EF
.text:00B143EF ; ---------------------------------------------------------------------------
.text:00B143F0                 align 20h
.text:00B14400
.text:00B14400 ; =============== S U B R O U T I N E =======================================
.text:00B14400
.text:00B14400 ; Attributes: bp-based frame
.text:00B14400
.text:00B14400 sub_B14400      proc near               ; CODE XREF: sub_B1125D↑j
.text:00B14400
.text:00B14400 var_4           = dword ptr -4
.text:00B14400
.text:00B14400                 push    ebp
.text:00B14401                 mov     ebp, esp
.text:00B14403                 push    ecx
.text:00B14404                 cmp     dword_B1C028, 0
.text:00B1440B                 jnz     short loc_B14416
.text:00B1440D                 mov     [ebp+var_4], 1
.text:00B14414                 jmp     short loc_B1441D
.text:00B14416 ; ---------------------------------------------------------------------------
.text:00B14416
.text:00B14416 loc_B14416:                             ; CODE XREF: sub_B14400+B↑j
.text:00B14416                 mov     [ebp+var_4], 0
.text:00B1441D
.text:00B1441D loc_B1441D:                             ; CODE XREF: sub_B14400+14↑j
.text:00B1441D                 mov     eax, [ebp+var_4]
.text:00B14420                 mov     esp, ebp
.text:00B14422                 pop     ebp
.text:00B14423                 retn
.text:00B14423 sub_B14400      endp
.text:00B14423
.text:00B14423 ; ---------------------------------------------------------------------------
.text:00B14424                 align 10h
.text:00B14430
.text:00B14430 ; =============== S U B R O U T I N E =======================================
.text:00B14430
.text:00B14430 ; Attributes: bp-based frame
.text:00B14430
.text:00B14430 sub_B14430      proc near               ; CODE XREF: sub_B1105A↑j
.text:00B14430                 push    ebp
.text:00B14431                 mov     ebp, esp
.text:00B14433                 mov     eax, offset unk_B1C58C
.text:00B14438                 pop     ebp
.text:00B14439                 retn
.text:00B14439 sub_B14430      endp
.text:00B14439
.text:00B14439 ; ---------------------------------------------------------------------------
.text:00B1443A                 align 10h
.text:00B14440
```

```
.text:00B14440 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14440
.text:00B14440 ; Attributes: bp-based frame
.text:00B14440
.text:00B14440 sub_B14440      proc near               ; CODE XREF: sub_B11037↑j
.text:00B14440                 push    ebp
.text:00B14441                 mov     ebp, esp
.text:00B14443                 mov     eax, offset unk_B1C580
.text:00B14448                 pop     ebp
.text:00B14449                 retn
.text:00B14449 sub_B14440      endp
.text:00B14449
.text:00B14449 ; ───────────────────────────────────────────────────────────────────────────
.text:00B1444A                 align 10h
.text:00B14450
.text:00B14450 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14450
.text:00B14450 ; Attributes: bp-based frame
.text:00B14450
.text:00B14450 sub_B14450      proc near               ; CODE XREF: sub_B1121C↑j
.text:00B14450
.text:00B14450 var_32C         = dword ptr -32Ch
.text:00B14450 var_2A0         = word ptr -2A0h
.text:00B14450 var_29C         = word ptr -29Ch
.text:00B14450 var_298         = word ptr -298h
.text:00B14450 var_294         = word ptr -294h
.text:00B14450 var_290         = dword ptr -290h
.text:00B14450 var_28C         = dword ptr -28Ch
.text:00B14450 var_288         = dword ptr -288h
.text:00B14450 var_284         = dword ptr -284h
.text:00B14450 var_280         = dword ptr -280h
.text:00B14450 var_27C         = dword ptr -27Ch
.text:00B14450 var_278         = dword ptr -278h
.text:00B14450 var_274         = dword ptr -274h
.text:00B14450 var_270         = word ptr -270h
.text:00B14450 var_26C         = dword ptr -26Ch
.text:00B14450 var_268         = dword ptr -268h
.text:00B14450 var_264         = word ptr -264h
.text:00B14450 var_60          = dword ptr -60h
.text:00B14450 var_5C          = dword ptr -5Ch
.text:00B14450 var_54          = dword ptr -54h
.text:00B14450 ExceptionInfo   = _EXCEPTION_POINTERS ptr -10h
.text:00B14450 var_8           = dword ptr -8
.text:00B14450 var_2           = byte ptr -2
.text:00B14450 var_1           = byte ptr -1
.text:00B14450 arg_0           = dword ptr  8
.text:00B14450
.text:00B14450                 push    ebp
.text:00B14451                 mov     ebp, esp
.text:00B14453                 sub     esp, 32Ch
.text:00B14459                 push    17h             ; ProcessorFeature
.text:00B1445B                 call    ds:IsProcessorFeaturePresent
.text:00B14461                 test    eax, eax
.text:00B14463                 jz      short loc_B1446A
.text:00B14465                 mov     ecx, [ebp+arg_0]
.text:00B14468                 int     29h             ; Win8: RtlFailFast(ecx)
.text:00B1446A ; ───────────────────────────────────────────────────────────────────────────
.text:00B1446A
.text:00B1446A loc_B1446A:                             ; CODE XREF: sub_B14450+13↑j
.text:00B1446A                 push    3
.text:00B1446C                 call    sub_B11302
.text:00B14471                 add     esp, 4
.text:00B14474                 push    2CCh            ; Size
.text:00B14479                 push    0               ; Val
.text:00B1447B                 lea     eax, [ebp+var_32C]
.text:00B14481                 push    eax             ; void *
.text:00B14482                 call    j_memset
.text:00B14487                 add     esp, 0Ch
.text:00B1448A                 mov     [ebp+var_27C], eax
.text:00B14490                 mov     [ebp+var_280], ecx
.text:00B14496                 mov     [ebp+var_284], edx
.text:00B1449C                 mov     [ebp+var_288], ebx
.text:00B144A2                 mov     [ebp+var_28C], esi
.text:00B144A8                 mov     [ebp+var_290], edi
.text:00B144AE                 mov     [ebp+var_264], ss
```

```
.text:00B144B5                 mov     [ebp+var_270], cs
.text:00B144BC                 mov     [ebp+var_294], ds
.text:00B144C3                 mov     [ebp+var_298], es
.text:00B144CA                 mov     [ebp+var_29C], fs
.text:00B144D1                 mov     [ebp+var_2A0], gs
.text:00B144D8                 pushf
.text:00B144D9                 pop     [ebp+var_26C]
.text:00B144DF                 mov     [ebp+var_32C], 10001h
.text:00B144E9                 mov     ecx, [ebp+4]
.text:00B144EC                 mov     [ebp+var_274], ecx
.text:00B144F2                 lea     edx, [ebp+4]
.text:00B144F5                 mov     [ebp+var_268], edx
.text:00B144FB                 lea     eax, [ebp+4]
.text:00B144FE                 mov     ecx, [eax-4]
.text:00B14501                 mov     [ebp+var_278], ecx
.text:00B14507                 push    50h ; 'P'       ; Size
.text:00B14509                 push    0               ; Val
.text:00B1450B                 lea     edx, [ebp+var_60]
.text:00B1450E                 push    edx             ; void *
.text:00B1450F                 call    j_memset
.text:00B14514                 add     esp, 0Ch
.text:00B14517                 mov     [ebp+var_60], 40000015h
.text:00B1451E                 mov     [ebp+var_5C], 1
.text:00B14525                 mov     eax, [ebp+4]
.text:00B14528                 mov     [ebp+var_54], eax
.text:00B1452B                 call    ds:IsDebuggerPresent
.text:00B14531                 cmp     eax, 1
.text:00B14534                 jnz     short loc_B1453C
.text:00B14536                 mov     [ebp+var_1], 1
.text:00B1453A                 jmp     short loc_B14540
.text:00B1453C ; ---------------------------------------------------------------------------
.text:00B1453C
.text:00B1453C loc_B1453C:                             ; CODE XREF: sub_B14450+E4↑j
.text:00B1453C                 mov     [ebp+var_1], 0
.text:00B14540
.text:00B14540 loc_B14540:                             ; CODE XREF: sub_B14450+EA↑j
.text:00B14540                 mov     cl, [ebp+var_1]
.text:00B14543                 mov     [ebp+var_2], cl
.text:00B14546                 lea     edx, [ebp+var_60]
.text:00B14549                 mov     [ebp+ExceptionInfo.ExceptionRecord], edx
.text:00B1454C                 lea     eax, [ebp+var_32C]
.text:00B14552                 mov     [ebp+ExceptionInfo.ContextRecord], eax
.text:00B14555                 push    0               ; lpTopLevelExceptionFilter
.text:00B14557                 call    ds:SetUnhandledExceptionFilter
.text:00B1455D                 lea     ecx, [ebp+ExceptionInfo]
.text:00B14560                 push    ecx             ; ExceptionInfo
.text:00B14561                 call    ds:UnhandledExceptionFilter
.text:00B14567                 mov     [ebp+var_8], eax
.text:00B1456A                 cmp     [ebp+var_8], 0
.text:00B1456E                 jnz     short loc_B14582
.text:00B14570                 movzx   edx, [ebp+var_2]
.text:00B14574                 test    edx, edx
.text:00B14576                 jnz     short loc_B14582
.text:00B14578                 push    3
.text:00B1457A                 call    sub_B11302
.text:00B1457F                 add     esp, 4
.text:00B14582
.text:00B14582 loc_B14582:                             ; CODE XREF: sub_B14450+11E↑j
.text:00B14582                                         ; sub_B14450+126↑j
.text:00B14582                 mov     esp, ebp
.text:00B14584                 pop     ebp
.text:00B14585                 retn
.text:00B14585 sub_B14450      endp
.text:00B14585
.text:00B14585 ; ---------------------------------------------------------------------------
.text:00B14586                 db 5Ah dup(0CCh)
.text:00B145E0 ; ---------------------------------------------------------------------------
.text:00B145E0
.text:00B145E0 loc_B145E0:                             ; CODE XREF: .text:00B110E1↑j
.text:00B145E0                 push    ebp
.text:00B145E1                 mov     ebp, esp
.text:00B145E3                 sub     esp, 48h
.text:00B145E6                 push    44h ; 'D'
.text:00B145E8                 push    0
.text:00B145EA                 lea     eax, [ebp-48h]
```

```
.text:00B145ED                 push    eax
.text:00B145EE                 call    j_memset
.text:00B145F3                 add     esp, 0Ch
.text:00B145F6                 lea     ecx, [ebp-48h]
.text:00B145F9                 push    ecx
.text:00B145FA                 call    ds:GetStartupInfoW
.text:00B14600                 mov     edx, [ebp-1Ch]
.text:00B14603                 and     edx, 1
.text:00B14606                 jz      short loc_B14611
.text:00B14608                 movzx   eax, word ptr [ebp-18h]
.text:00B1460C                 mov     [ebp-4], eax
.text:00B1460F                 jmp     short loc_B14618
.text:00B14611 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14611
.text:00B14611 loc_B14611:                             ; CODE XREF: .text:00B14606↑j
.text:00B14611                 mov     dword ptr [ebp-4], 0Ah
.text:00B14618
.text:00B14618 loc_B14618:                             ; CODE XREF: .text:00B1460F↑j
.text:00B14618                 movzx   eax, word ptr [ebp-4]
.text:00B1461C                 mov     esp, ebp
.text:00B1461E                 pop     ebp
.text:00B1461F                 retn
.text:00B1461F ; ───────────────────────────────────────────────────────────────────────────
.text:00B14620                 db 10h dup(0CCh)
.text:00B14630
.text:00B14630 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14630
.text:00B14630 ; Attributes: bp-based frame
.text:00B14630
.text:00B14630 sub_B14630      proc near               ; CODE XREF: sub_B11267↑j
.text:00B14630                 push    ebp
.text:00B14631                 mov     ebp, esp
.text:00B14633                 call    sub_B11064
.text:00B14638                 pop     ebp
.text:00B14639                 retn
.text:00B14639 sub_B14630      endp
.text:00B14639
.text:00B14639 ; ───────────────────────────────────────────────────────────────────────────
.text:00B1463A                 align 10h
.text:00B14640
.text:00B14640 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14640
.text:00B14640 ; Attributes: bp-based frame
.text:00B14640
.text:00B14640 sub_B14640      proc near               ; CODE XREF: sub_B112D5↑j
.text:00B14640                 push    ebp
.text:00B14641                 mov     ebp, esp
.text:00B14643                 xor     eax, eax
.text:00B14645                 pop     ebp
.text:00B14646                 retn
.text:00B14646 sub_B14640      endp
.text:00B14646
.text:00B14646 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14647                 align 10h
.text:00B14650
.text:00B14650 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14650
.text:00B14650 ; Attributes: bp-based frame
.text:00B14650
.text:00B14650 sub_B14650      proc near               ; CODE XREF: sub_B11357↑j
.text:00B14650
.text:00B14650 var_8           = dword ptr -8
.text:00B14650 var_4           = dword ptr -4
.text:00B14650
.text:00B14650                 push    ebp
.text:00B14651                 mov     ebp, esp
.text:00B14653                 sub     esp, 8
.text:00B14656                 push    0               ; lpModuleName
.text:00B14658                 call    ds:GetModuleHandleW
.text:00B1465E                 mov     [ebp+var_4], eax
.text:00B14661                 cmp     [ebp+var_4], 0
.text:00B14665                 jnz     short loc_B1466B
.text:00B14667                 xor     al, al
.text:00B14669                 jmp     short loc_B146CF
.text:00B1466B ; ───────────────────────────────────────────────────────────────────────────
```

```
.text:00B1466B
.text:00B1466B loc_B1466B:                             ; CODE XREF: sub_B14650+15↑j
.text:00B1466B                 mov     eax, [ebp+var_4]
.text:00B1466E                 movzx   ecx, word ptr [eax]
.text:00B14671                 cmp     ecx, 5A4Dh
.text:00B14677                 jz      short loc_B1467D
.text:00B14679                 xor     al, al
.text:00B1467B                 jmp     short loc_B146CF
.text:00B1467D ; ───────────────────────────────────────────────────────────────────────────
.text:00B1467D
.text:00B1467D loc_B1467D:                             ; CODE XREF: sub_B14650+27↑j
.text:00B1467D                 mov     edx, [ebp+var_4]
.text:00B14680                 mov     eax, [ebp+var_4]
.text:00B14683                 add     eax, [edx+3Ch]
.text:00B14686                 mov     [ebp+var_8], eax
.text:00B14689                 mov     ecx, [ebp+var_8]
.text:00B1468C                 cmp     dword ptr [ecx], 4550h
.text:00B14692                 jz      short loc_B14698
.text:00B14694                 xor     al, al
.text:00B14696                 jmp     short loc_B146CF
.text:00B14698 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14698
.text:00B14698 loc_B14698:                             ; CODE XREF: sub_B14650+42↑j
.text:00B14698                 mov     edx, [ebp+var_8]
.text:00B1469B                 movzx   eax, word ptr [edx+18h]
.text:00B1469F                 cmp     eax, 10Bh
.text:00B146A4                 jz      short loc_B146AA
.text:00B146A6                 xor     al, al
.text:00B146A8                 jmp     short loc_B146CF
.text:00B146AA ; ───────────────────────────────────────────────────────────────────────────
.text:00B146AA
.text:00B146AA loc_B146AA:                             ; CODE XREF: sub_B14650+54↑j
.text:00B146AA                 mov     ecx, [ebp+var_8]
.text:00B146AD                 cmp     dword ptr [ecx+74h], 0Eh
.text:00B146B1                 ja      short loc_B146B7
.text:00B146B3                 xor     al, al
.text:00B146B5                 jmp     short loc_B146CF
.text:00B146B7 ; ───────────────────────────────────────────────────────────────────────────
.text:00B146B7
.text:00B146B7 loc_B146B7:                             ; CODE XREF: sub_B14650+61↑j
.text:00B146B7                 mov     edx, 8
.text:00B146BC                 imul    eax, edx, 0Eh
.text:00B146BF                 mov     ecx, [ebp+var_8]
.text:00B146C2                 cmp     dword ptr [ecx+eax+78h], 0
.text:00B146C7                 jnz     short loc_B146CD
.text:00B146C9                 xor     al, al
.text:00B146CB                 jmp     short loc_B146CF
.text:00B146CD ; ───────────────────────────────────────────────────────────────────────────
.text:00B146CD
.text:00B146CD loc_B146CD:                             ; CODE XREF: sub_B14650+77↑j
.text:00B146CD                 mov     al, 1
.text:00B146CF
.text:00B146CF loc_B146CF:                             ; CODE XREF: sub_B14650+19↑j
.text:00B146CF                                         ; sub_B14650+2B↑j ...
.text:00B146CF                 mov     esp, ebp
.text:00B146D1                 pop     ebp
.text:00B146D2                 retn
.text:00B146D2 sub_B14650      endp
.text:00B146D2
.text:00B146D2 ; ───────────────────────────────────────────────────────────────────────────
.text:00B146D3                 align 40h
.text:00B14700
.text:00B14700 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14700
.text:00B14700 ; Attributes: bp-based frame
.text:00B14700
.text:00B14700 sub_B14700      proc near               ; CODE XREF: sub_B1138E↑j
.text:00B14700                 push    ebp
.text:00B14701                 mov     ebp, esp
.text:00B14703                 push    offset TopLevelExceptionFilter ; lpTopLevelExceptionFilter
.text:00B14708                 call    ds:SetUnhandledExceptionFilter
.text:00B1470E                 pop     ebp
.text:00B1470F                 retn
.text:00B1470F sub_B14700      endp
.text:00B1470F
```

```
.text:00B1470F ; ───────────────────────────────────────────────────────────────────────────
.text:00B14710                 align 20h
.text:00B14720
.text:00B14720 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14720
.text:00B14720 ; Attributes: bp-based frame
.text:00B14720
.text:00B14720 sub_B14720      proc near               ; CODE XREF: sub_B11064↑j
.text:00B14720                                         ; .text:00B11271↑j
.text:00B14720                 push    ebp
.text:00B14721                 mov     ebp, esp
.text:00B14723                 xor     eax, eax
.text:00B14725                 pop     ebp
.text:00B14726                 retn
.text:00B14726 sub_B14720      endp
.text:00B14726
.text:00B14726 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14727                 align 10h
.text:00B14730
.text:00B14730 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14730
.text:00B14730 ; Attributes: bp-based frame
.text:00B14730
.text:00B14730 TopLevelExceptionFilter_0 proc near     ; CODE XREF: TopLevelExceptionFilter↑j
.text:00B14730
.text:00B14730 var_8           = dword ptr -8
.text:00B14730 var_4           = dword ptr -4
.text:00B14730 arg_0           = dword ptr  8
.text:00B14730
.text:00B14730                 push    ebp
.text:00B14731                 mov     ebp, esp
.text:00B14733                 sub     esp, 8
.text:00B14736                 mov     eax, [ebp+arg_0]
.text:00B14739                 mov     ecx, [eax]
.text:00B1473B                 mov     [ebp+var_4], ecx
.text:00B1473E                 mov     edx, [ebp+var_4]
.text:00B14741                 cmp     dword ptr [edx], 0E06D7363h
.text:00B14747                 jnz     short loc_B147A4
.text:00B14749                 mov     eax, [ebp+var_4]
.text:00B1474C                 cmp     dword ptr [eax+10h], 3
.text:00B14750                 jnz     short loc_B147A4
.text:00B14752                 mov     ecx, [ebp+var_4]
.text:00B14755                 cmp     dword ptr [ecx+14h], 19930520h
.text:00B1475C                 jz      short loc_B14782
.text:00B1475E                 mov     edx, [ebp+var_4]
.text:00B14761                 cmp     dword ptr [edx+14h], 19930521h
.text:00B14768                 jz      short loc_B14782
.text:00B1476A                 mov     eax, [ebp+var_4]
.text:00B1476D                 cmp     dword ptr [eax+14h], 19930522h
.text:00B14774                 jz      short loc_B14782
.text:00B14776                 mov     ecx, [ebp+var_4]
.text:00B14779                 cmp     dword ptr [ecx+14h], 1994000h
.text:00B14780                 jnz     short loc_B147A4
.text:00B14782
.text:00B14782 loc_B14782:                             ; CODE XREF: TopLevelExceptionFilter_0+2C↑j
.text:00B14782                                         ; TopLevelExceptionFilter_0+38↑j  ...
.text:00B14782                 call    j___current_exception
.text:00B14787                 mov     edx, [ebp+var_4]
.text:00B1478A                 mov     [eax], edx
.text:00B1478C                 mov     eax, [ebp+arg_0]
.text:00B1478F                 mov     ecx, [eax+4]
.text:00B14792                 mov     [ebp+var_8], ecx
.text:00B14795                 call    j___current_exception_context
.text:00B1479A                 mov     edx, [ebp+var_8]
.text:00B1479D                 mov     [eax], edx
.text:00B1479F                 call    j_terminate
.text:00B147A4
.text:00B147A4 loc_B147A4:                             ; CODE XREF: TopLevelExceptionFilter_0+17↑j
.text:00B147A4                                         ; TopLevelExceptionFilter_0+20↑j  ...
.text:00B147A4                 xor     eax, eax
.text:00B147A6                 mov     esp, ebp
.text:00B147A8                 pop     ebp
.text:00B147A9                 retn    4
.text:00B147A9 TopLevelExceptionFilter_0 endp
.text:00B147A9
```

```
.text:00B147A9 ; ───────────────────────────────────────────────────────────────────────────
.text:00B147AC                 db 24h dup(0CCh)
.text:00B147D0
.text:00B147D0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B147D0
.text:00B147D0 ; Attributes: bp-based frame
.text:00B147D0
.text:00B147D0 sub_B147D0      proc near               ; CODE XREF: sub_B11302↑j
.text:00B147D0                 push    ebp
.text:00B147D1                 mov     ebp, esp
.text:00B147D3                 mov     dword_B1C564, 0
.text:00B147DD                 pop     ebp
.text:00B147DE                 retn
.text:00B147DE sub_B147D0      endp
.text:00B147DE
.text:00B147DE ; ───────────────────────────────────────────────────────────────────────────
.text:00B147DF                 db 11h dup(0CCh)
.text:00B147F0
.text:00B147F0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B147F0
.text:00B147F0
.text:00B147F0 sub_B147F0      proc near               ; CODE XREF: sub_B1100A↑j
.text:00B147F0                 push    esi
.text:00B147F1                 mov     esi, offset unk_B1ABDC
.text:00B147F6                 mov     eax, esi
.text:00B147F8                 cmp     eax, offset unk_B1ADE0
.text:00B147FD                 jnb     short loc_B1481C
.text:00B147FF                 push    edi
.text:00B14800
.text:00B14800 loc_B14800:                             ; CODE XREF: sub_B147F0+29↓j
.text:00B14800                 mov     edi, [esi]
.text:00B14802                 test    edi, edi
.text:00B14804                 jz      short loc_B14810
.text:00B14806                 mov     ecx, edi
.text:00B14808                 call    ds:___guard_check_icall_fptr
.text:00B1480E                 call    edi
.text:00B14810
.text:00B14810 loc_B14810:                             ; CODE XREF: sub_B147F0+14↑j
.text:00B14810                 add     esi, 4
.text:00B14813                 cmp     esi, offset unk_B1ADE0
.text:00B14819                 jb      short loc_B14800
.text:00B1481B                 pop     edi
.text:00B1481C
.text:00B1481C loc_B1481C:                             ; CODE XREF: sub_B147F0+D↑j
.text:00B1481C                 pop     esi
.text:00B1481D                 retn
.text:00B1481D sub_B147F0      endp
.text:00B1481D
.text:00B1481D ; ───────────────────────────────────────────────────────────────────────────
.text:00B1481E                 db 12h dup(0CCh)
.text:00B14830
.text:00B14830 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14830
.text:00B14830
.text:00B14830 sub_B14830      proc near               ; CODE XREF: Function↑j
.text:00B14830                 push    esi
.text:00B14831                 mov     esi, offset unk_B1AEE8
.text:00B14836                 mov     eax, esi
.text:00B14838                 cmp     eax, offset unk_B1B0EC
.text:00B1483D                 jnb     short loc_B1485C
.text:00B1483F                 push    edi
.text:00B14840
.text:00B14840 loc_B14840:                             ; CODE XREF: sub_B14830+29↓j
.text:00B14840                 mov     edi, [esi]
.text:00B14842                 test    edi, edi
.text:00B14844                 jz      short loc_B14850
.text:00B14846                 mov     ecx, edi
.text:00B14848                 call    ds:___guard_check_icall_fptr
.text:00B1484E                 call    edi
.text:00B14850
.text:00B14850 loc_B14850:                             ; CODE XREF: sub_B14830+14↑j
.text:00B14850                 add     esi, 4
.text:00B14853                 cmp     esi, offset unk_B1B0EC
.text:00B14859                 jb      short loc_B14840
.text:00B1485B                 pop     edi
```

```
.text:00B1485C
.text:00B1485C loc_B1485C:                             ; CODE XREF: sub_B14830+D↑j
.text:00B1485C                 pop     esi
.text:00B1485D                 retn
.text:00B1485D sub_B14830      endp
.text:00B1485D
.text:00B1485D ; ───────────────────────────────────────────────────────────────────────────
.text:00B1485E                 db 12h dup(0CCh)
.text:00B14870
.text:00B14870 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14870
.text:00B14870 ; Attributes: bp-based frame
.text:00B14870
.text:00B14870 SEH_4128D0      proc near               ; CODE XREF: SEH_4132B0+14↓p
.text:00B14870                                         ; DATA XREF: sub_B128D0+A↑o ...
.text:00B14870
.text:00B14870 arg_0           = dword ptr  8
.text:00B14870 arg_4           = dword ptr  0Ch
.text:00B14870 arg_8           = dword ptr  10h
.text:00B14870 arg_C           = dword ptr  14h
.text:00B14870
.text:00B14870 SEH_413E60:
.text:00B14870                 push    ebp
.text:00B14871                 mov     ebp, esp
.text:00B14873                 mov     eax, [ebp+arg_0]
.text:00B14876                 mov     ecx, [eax]
.text:00B14878                 push    ecx
.text:00B14879                 call    sub_B1140B
.text:00B1487E                 add     esp, 4
.text:00B14881                 mov     edx, [ebp+arg_0]
.text:00B14884                 mov     [edx], eax
.text:00B14886                 mov     eax, [ebp+arg_C]
.text:00B14889                 push    eax
.text:00B1488A                 mov     ecx, [ebp+arg_8]
.text:00B1488D                 push    ecx
.text:00B1488E                 mov     edx, [ebp+arg_4]
.text:00B14891                 push    edx
.text:00B14892                 mov     eax, [ebp+arg_0]
.text:00B14895                 push    eax
.text:00B14896                 push    offset j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B1489B                 push    offset ___security_cookie
.text:00B148A0                 call    j__except_handler4_common
.text:00B148A5                 add     esp, 18h
.text:00B148A8                 pop     ebp
.text:00B148A9                 retn
.text:00B148A9 SEH_4128D0      endp
.text:00B148A9
.text:00B148A9 ; ───────────────────────────────────────────────────────────────────────────
.text:00B148AA                 align 20h
.text:00B148C0
.text:00B148C0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B148C0
.text:00B148C0 ; Attributes: bp-based frame
.text:00B148C0
.text:00B148C0 sub_B148C0      proc near               ; CODE XREF: sub_B112CB↑j
.text:00B148C0
.text:00B148C0 var_4           = dword ptr -4
.text:00B148C0
.text:00B148C0                 push    ebp
.text:00B148C1                 mov     ebp, esp
.text:00B148C3                 push    ecx
.text:00B148C4                 mov     [ebp+var_4], ecx
.text:00B148C7                 mov     esp, ebp
.text:00B148C9                 pop     ebp
.text:00B148CA                 retn
.text:00B148CA sub_B148C0      endp
.text:00B148CA
.text:00B148CA ; ───────────────────────────────────────────────────────────────────────────
.text:00B148CB                 align 10h
.text:00B148D0
.text:00B148D0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B148D0
.text:00B148D0 ; Attributes: bp-based frame
.text:00B148D0
.text:00B148D0 sub_B148D0      proc near               ; CODE XREF: sub_B1137A↑j
```

```
.text:00B148D0
.text:00B148D0 var_4           = dword ptr -4
.text:00B148D0 arg_0           = dword ptr  8
.text:00B148D0
.text:00B148D0                 push    ebp
.text:00B148D1                 mov     ebp, esp
.text:00B148D3                 push    ecx
.text:00B148D4                 mov     eax, [ebp+arg_0]
.text:00B148D7                 mov     ecx, [eax]
.text:00B148D9                 mov     [ebp+var_4], ecx
.text:00B148DC                 mov     eax, [ebp+var_4]
.text:00B148DF                 mov     esp, ebp
.text:00B148E1                 pop     ebp
.text:00B148E2                 retn
.text:00B148E2 sub_B148D0      endp
.text:00B148E2
.text:00B148E2 ; ───────────────────────────────────────────────────────────────────────────
.text:00B148E3                 align 10h
.text:00B148F0
.text:00B148F0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B148F0
.text:00B148F0 ; Attributes: bp-based frame
.text:00B148F0
.text:00B148F0 sub_B148F0      proc near               ; CODE XREF: sub_B110B9↑j
.text:00B148F0
.text:00B148F0 arg_0           = dword ptr  8
.text:00B148F0
.text:00B148F0                 push    ebp
.text:00B148F1                 mov     ebp, esp
.text:00B148F3                 mov     eax, [ebp+arg_0]
.text:00B148F6                 push    eax
.text:00B148F7                 call    sub_B1137A
.text:00B148FC                 add     esp, 4
.text:00B148FF                 pop     ebp
.text:00B14900                 retn
.text:00B14900 sub_B148F0      endp
.text:00B14900
.text:00B14900 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14901                 align 10h
.text:00B14910
.text:00B14910 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14910
.text:00B14910 ; Attributes: bp-based frame
.text:00B14910
.text:00B14910 sub_B14910      proc near               ; CODE XREF: sub_B1124E↑j
.text:00B14910
.text:00B14910 arg_0           = dword ptr  8
.text:00B14910 arg_4           = dword ptr  0Ch
.text:00B14910
.text:00B14910                 push    ebp
.text:00B14911                 mov     ebp, esp
.text:00B14913                 mov     eax, [ebp+arg_4]
.text:00B14916                 mov     ecx, [ebp+arg_0]
.text:00B14919                 xchg    eax, [ecx]
.text:00B1491B                 pop     ebp
.text:00B1491C                 retn
.text:00B1491C sub_B14910      endp
.text:00B1491C
.text:00B1491C ; ───────────────────────────────────────────────────────────────────────────
.text:00B1491D                 align 10h
.text:00B14920
.text:00B14920 loc_B14920:                             ; CODE XREF: .text:00B112F8↑j
.text:00B14920                 push    ebp
.text:00B14921                 mov     ebp, esp
.text:00B14923                 mov     eax, [ebp+8]
.text:00B14926                 push    eax
.text:00B14927                 call    sub_B14A50
.text:00B1492C                 add     esp, 4
.text:00B1492F                 test    eax, eax
.text:00B14931                 jz      short loc_B14934
.text:00B14933                 int     3               ; Trap to Debugger
.text:00B14934
.text:00B14934 loc_B14934:                             ; CODE XREF: .text:00B14931↑j
.text:00B14934                 pop     ebp
.text:00B14935                 retn
```

```
.text:00B14935 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14936                 align 10h
.text:00B14940
.text:00B14940 loc_B14940:                             ; CODE XREF: .text:00B11244↑j
.text:00B14940                 push    ebp
.text:00B14941                 mov     ebp, esp
.text:00B14943                 mov     eax, [ebp+8]
.text:00B14946                 push    eax
.text:00B14947                 call    sub_B14A50
.text:00B1494C                 add     esp, 4
.text:00B1494F                 test    eax, eax
.text:00B14951                 jz      short loc_B1495A
.text:00B14953                 mov     ecx, 41h ; 'A'
.text:00B14958                 int     29h             ; Win8: RtlFailFast(ecx)
.text:00B1495A ; ───────────────────────────────────────────────────────────────────────────
.text:00B1495A
.text:00B1495A loc_B1495A:                             ; CODE XREF: .text:00B14951↑j
.text:00B1495A                 pop     ebp
.text:00B1495B                 retn
.text:00B1495B ; ───────────────────────────────────────────────────────────────────────────
.text:00B1495C                 db 14h dup(0CCh)
.text:00B14970 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14970
.text:00B14970 loc_B14970:                             ; CODE XREF: .text:00B11087↑j
.text:00B14970                 push    ebp
.text:00B14971                 mov     ebp, esp
.text:00B14973                 pop     ebp
.text:00B14974                 retn
.text:00B14974 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14975                 align 10h
.text:00B14980
.text:00B14980 loc_B14980:                             ; CODE XREF: .text:00B11316↑j
.text:00B14980                 push    ebp
.text:00B14981                 mov     ebp, esp
.text:00B14983                 mov     eax, [ebp+8]
.text:00B14986                 push    eax
.text:00B14987                 call    sub_B14A50
.text:00B1498C                 add     esp, 4
.text:00B1498F                 test    eax, eax
.text:00B14991                 jz      short loc_B1499F
.text:00B14993                 mov     ecx, [ebp+8]
.text:00B14996                 push    ecx
.text:00B14997                 call    sub_B149B0
.text:00B1499C                 add     esp, 4
.text:00B1499F
.text:00B1499F loc_B1499F:                             ; CODE XREF: .text:00B14991↑j
.text:00B1499F                 pop     ebp
.text:00B149A0                 retn
.text:00B149A0 ; ───────────────────────────────────────────────────────────────────────────
.text:00B149A1                 align 10h
.text:00B149B0
.text:00B149B0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B149B0
.text:00B149B0 ; Attributes: bp-based frame
.text:00B149B0
.text:00B149B0 sub_B149B0      proc near               ; CODE XREF: .text:00B14997↑p
.text:00B149B0                                         ; .text:00B14B9F↓p
.text:00B149B0
.text:00B149B0 arg_0           = dword ptr  8
.text:00B149B0
.text:00B149B0                 push    ebp
.text:00B149B1                 mov     ebp, esp
.text:00B149B3                 cmp     ds:___castguard_check_failure_os_handled_fptr, 0
.text:00B149BA                 jz      short loc_B149CB
.text:00B149BC                 mov     eax, [ebp+arg_0]
.text:00B149BF                 push    eax
.text:00B149C0                 mov     ecx, ds:___castguard_check_failure_os_handled_fptr
.text:00B149C6                 call    ecx ; ___castguard_check_failure_os_handled_fptr
.text:00B149C8                 add     esp, 4
.text:00B149CB
.text:00B149CB loc_B149CB:                             ; CODE XREF: sub_B149B0+A↑j
.text:00B149CB                 pop     ebp
.text:00B149CC                 retn
.text:00B149CC sub_B149B0      endp
.text:00B149CC
```

```
.text:00B149CC ; ---------------------------------------------------------------------------
.text:00B149CD                 align 20h
.text:00B149E0
.text:00B149E0 loc_B149E0:                             ; CODE XREF: .text:00B11145↑j
.text:00B149E0                 push    ebp
.text:00B149E1                 mov     ebp, esp
.text:00B149E3                 mov     eax, [ebp+8]
.text:00B149E6                 push    eax
.text:00B149E7                 call    sub_B14A50
.text:00B149EC                 add     esp, 4
.text:00B149EF                 test    eax, eax
.text:00B149F1                 jz      short loc_B149FF
.text:00B149F3                 mov     ecx, [ebp+8]
.text:00B149F6                 push    ecx
.text:00B149F7                 call    sub_B14A10
.text:00B149FC                 add     esp, 4
.text:00B149FF
.text:00B149FF loc_B149FF:                             ; CODE XREF: .text:00B149F1↑j
.text:00B149FF                 pop     ebp
.text:00B14A00                 retn
.text:00B14A00 ; ---------------------------------------------------------------------------
.text:00B14A01                 align 10h
.text:00B14A10
.text:00B14A10 ; =============== S U B R O U T I N E =======================================
.text:00B14A10
.text:00B14A10 ; Attributes: bp-based frame
.text:00B14A10
.text:00B14A10 sub_B14A10      proc near               ; CODE XREF: .text:00B149F7↑p
.text:00B14A10                                         ; .text:00B14BEF↓p
.text:00B14A10
.text:00B14A10 var_8           = dword ptr -8
.text:00B14A10 var_4           = dword ptr -4
.text:00B14A10 arg_0           = dword ptr  8
.text:00B14A10
.text:00B14A10                 push    ebp
.text:00B14A11                 mov     ebp, esp
.text:00B14A13                 sub     esp, 8
.text:00B14A16                 mov     eax, dword_B1C568
.text:00B14A1B                 mov     [ebp+var_4], eax
.text:00B14A1E                 cmp     [ebp+var_4], 0
.text:00B14A22                 jz      short loc_B14A3D
.text:00B14A24                 mov     ecx, [ebp+arg_0]
.text:00B14A27                 push    ecx
.text:00B14A28                 mov     edx, [ebp+var_4]
.text:00B14A2B                 mov     [ebp+var_8], edx
.text:00B14A2E                 mov     ecx, [ebp+var_8]
.text:00B14A31                 call    ds:___guard_check_icall_fptr
.text:00B14A37                 call    [ebp+var_8]
.text:00B14A3A                 add     esp, 4
.text:00B14A3D
.text:00B14A3D loc_B14A3D:                             ; CODE XREF: sub_B14A10+12↑j
.text:00B14A3D                 mov     esp, ebp
.text:00B14A3F                 pop     ebp
.text:00B14A40                 retn
.text:00B14A40 sub_B14A10      endp
.text:00B14A40
.text:00B14A40 ; ---------------------------------------------------------------------------
.text:00B14A41                 align 10h
.text:00B14A50
.text:00B14A50 ; =============== S U B R O U T I N E =======================================
.text:00B14A50
.text:00B14A50 ; Attributes: bp-based frame
.text:00B14A50
.text:00B14A50 sub_B14A50      proc near               ; CODE XREF: .text:00B14927↑p
.text:00B14A50                                         ; .text:00B14947↑p ...
.text:00B14A50
.text:00B14A50 var_10          = dword ptr -10h
.text:00B14A50 var_C           = dword ptr -0Ch
.text:00B14A50 var_8           = dword ptr -8
.text:00B14A50 var_4           = dword ptr -4
.text:00B14A50 arg_0           = dword ptr  8
.text:00B14A50
.text:00B14A50                 push    ebp
.text:00B14A51                 mov     ebp, esp
.text:00B14A53                 sub     esp, 10h
```

```
.text:00B14A56                mov     [ebp+var_4], offset unk_B1A7C0
.text:00B14A5D                mov     [ebp+var_C], offset unk_B1A900
.text:00B14A64                mov     eax, [ebp+var_C]
.text:00B14A67                sub     eax, [ebp+var_4]
.text:00B14A6A                mov     [ebp+var_10], eax
.text:00B14A6D                mov     ecx, [ebp+arg_0]
.text:00B14A70                sub     ecx, [ebp+var_4]
.text:00B14A73                cmp     ecx, [ebp+var_10]
.text:00B14A76                ja      short loc_B14A81
.text:00B14A78                mov     [ebp+var_8], 1
.text:00B14A7F                jmp     short loc_B14A88
.text:00B14A81 ; ────────────────────────────────────────────────────────────────────────────
.text:00B14A81
.text:00B14A81 loc_B14A81:                            ; CODE XREF: sub_B14A50+26↑j
.text:00B14A81                mov     [ebp+var_8], 0
.text:00B14A88
.text:00B14A88 loc_B14A88:                            ; CODE XREF: sub_B14A50+2F↑j
.text:00B14A88                mov     eax, [ebp+var_8]
.text:00B14A8B                mov     esp, ebp
.text:00B14A8D                pop     ebp
.text:00B14A8E                retn
.text:00B14A8E sub_B14A50      endp
.text:00B14A8E
.text:00B14A8E ; ────────────────────────────────────────────────────────────────────────────
.text:00B14A8F                align 20h
.text:00B14AA0
.text:00B14AA0 loc_B14AA0:                            ; CODE XREF: .text:00B112B2↑j
.text:00B14AA0                push    ebp
.text:00B14AA1                mov     ebp, esp
.text:00B14AA3                push    ecx
.text:00B14AA4                mov     eax, [ebp+8]
.text:00B14AA7                push    eax
.text:00B14AA8                push    offset dword_B1C568
.text:00B14AAD                call    sub_B1124E
.text:00B14AB2                add     esp, 8
.text:00B14AB5                mov     [ebp-4], eax
.text:00B14AB8                mov     eax, [ebp-4]
.text:00B14ABB                mov     esp, ebp
.text:00B14ABD                pop     ebp
.text:00B14ABE                retn
.text:00B14ABE ; ────────────────────────────────────────────────────────────────────────────
.text:00B14ABF                db 11h dup(0CCh)
.text:00B14AD0 ; ────────────────────────────────────────────────────────────────────────────
.text:00B14AD0
.text:00B14AD0 loc_B14AD0:                            ; CODE XREF: .text:00B11311↑j
.text:00B14AD0                push    ebp
.text:00B14AD1                mov     ebp, esp
.text:00B14AD3                mov     eax, [ebp+10h]
.text:00B14AD6                push    eax
.text:00B14AD7                mov     ecx, [ebp+0Ch]
.text:00B14ADA                push    ecx
.text:00B14ADB                mov     edx, [ebp+8]
.text:00B14ADE                push    edx
.text:00B14ADF                call    sub_B14C10
.text:00B14AE4                add     esp, 0Ch
.text:00B14AE7                test    eax, eax
.text:00B14AE9                jz      short loc_B14AFC
.text:00B14AEB                mov     eax, [ebp+8]
.text:00B14AEE                push    eax
.text:00B14AEF                call    sub_B14A50
.text:00B14AF4                add     esp, 4
.text:00B14AF7                test    eax, eax
.text:00B14AF9                jz      short loc_B14AFC
.text:00B14AFB                int     3               ; Trap to Debugger
.text:00B14AFC
.text:00B14AFC loc_B14AFC:                            ; CODE XREF: .text:00B14AE9↑j
.text:00B14AFC                                        ; .text:00B14AF9↑j
.text:00B14AFC                pop     ebp
.text:00B14AFD                retn
.text:00B14AFD ; ────────────────────────────────────────────────────────────────────────────
.text:00B14AFE                db 12h dup(0CCh)
.text:00B14B10 ; ────────────────────────────────────────────────────────────────────────────
.text:00B14B10
.text:00B14B10 loc_B14B10:                            ; CODE XREF: .text:00B11320↑j
.text:00B14B10                push    ebp
```

```
.text:00B14B11                 mov     ebp, esp
.text:00B14B13                 mov     eax, [ebp+10h]
.text:00B14B16                 push    eax
.text:00B14B17                 mov     ecx, [ebp+0Ch]
.text:00B14B1A                 push    ecx
.text:00B14B1B                 mov     edx, [ebp+8]
.text:00B14B1E                 push    edx
.text:00B14B1F                 call    sub_B14C10
.text:00B14B24                 add     esp, 0Ch
.text:00B14B27                 test    eax, eax
.text:00B14B29                 jz      short loc_B14B42
.text:00B14B2B                 mov     eax, [ebp+8]
.text:00B14B2E                 push    eax
.text:00B14B2F                 call    sub_B14A50
.text:00B14B34                 add     esp, 4
.text:00B14B37                 test    eax, eax
.text:00B14B39                 jz      short loc_B14B42
.text:00B14B3B                 mov     ecx, 41h ; 'A'
.text:00B14B40                 int     29h             ; Win8: RtlFailFast(ecx)
.text:00B14B42 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14B42
.text:00B14B42 loc_B14B42:                             ; CODE XREF: .text:00B14B29↑j
.text:00B14B42                                         ; .text:00B14B39↑j
.text:00B14B42                 pop     ebp
.text:00B14B43                 retn
.text:00B14B43 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14B44                 align 20h
.text:00B14B60
.text:00B14B60 loc_B14B60:                             ; CODE XREF: .text:00B112E4↑j
.text:00B14B60                 push    ebp
.text:00B14B61                 mov     ebp, esp
.text:00B14B63                 pop     ebp
.text:00B14B64                 retn
.text:00B14B64 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14B65                 align 10h
.text:00B14B70
.text:00B14B70 loc_B14B70:                             ; CODE XREF: .text:00B113B6↑j
.text:00B14B70                 push    ebp
.text:00B14B71                 mov     ebp, esp
.text:00B14B73                 mov     eax, [ebp+10h]
.text:00B14B76                 push    eax
.text:00B14B77                 mov     ecx, [ebp+0Ch]
.text:00B14B7A                 push    ecx
.text:00B14B7B                 mov     edx, [ebp+8]
.text:00B14B7E                 push    edx
.text:00B14B7F                 call    sub_B14C10
.text:00B14B84                 add     esp, 0Ch
.text:00B14B87                 test    eax, eax
.text:00B14B89                 jz      short loc_B14BA7
.text:00B14B8B                 mov     eax, [ebp+8]
.text:00B14B8E                 push    eax
.text:00B14B8F                 call    sub_B14A50
.text:00B14B94                 add     esp, 4
.text:00B14B97                 test    eax, eax
.text:00B14B99                 jz      short loc_B14BA7
.text:00B14B9B                 mov     ecx, [ebp+8]
.text:00B14B9E                 push    ecx
.text:00B14B9F                 call    sub_B149B0
.text:00B14BA4                 add     esp, 4
.text:00B14BA7
.text:00B14BA7 loc_B14BA7:                             ; CODE XREF: .text:00B14B89↑j
.text:00B14BA7                                         ; .text:00B14B99↑j
.text:00B14BA7                 pop     ebp
.text:00B14BA8                 retn
.text:00B14BA8 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14BA9                 align 20h
.text:00B14BC0
.text:00B14BC0 loc_B14BC0:                             ; CODE XREF: .text:00B110B4↑j
.text:00B14BC0                 push    ebp
.text:00B14BC1                 mov     ebp, esp
.text:00B14BC3                 mov     eax, [ebp+10h]
.text:00B14BC6                 push    eax
.text:00B14BC7                 mov     ecx, [ebp+0Ch]
.text:00B14BCA                 push    ecx
.text:00B14BCB                 mov     edx, [ebp+8]
```

```
.text:00B14BCE                 push    edx
.text:00B14BCF                 call    sub_B14C10
.text:00B14BD4                 add     esp, 0Ch
.text:00B14BD7                 test    eax, eax
.text:00B14BD9                 jz      short loc_B14BF7
.text:00B14BDB                 mov     eax, [ebp+8]
.text:00B14BDE                 push    eax
.text:00B14BDF                 call    sub_B14A50
.text:00B14BE4                 add     esp, 4
.text:00B14BE7                 test    eax, eax
.text:00B14BE9                 jz      short loc_B14BF7
.text:00B14BEB                 mov     ecx, [ebp+8]
.text:00B14BEE                 push    ecx
.text:00B14BEF                 call    sub_B14A10
.text:00B14BF4                 add     esp, 4
.text:00B14BF7
.text:00B14BF7 loc_B14BF7:                             ; CODE XREF: .text:00B14BD9↑j
.text:00B14BF7                                         ; .text:00B14BE9↑j
.text:00B14BF7                 pop     ebp
.text:00B14BF8                 retn
.text:00B14BF8 ; ────────────────────────────────────────────────────────────────────────────
.text:00B14BF9                 db 17h dup(0CCh)
.text:00B14C10
.text:00B14C10 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14C10
.text:00B14C10 ; Attributes: bp-based frame
.text:00B14C10
.text:00B14C10 sub_B14C10      proc near               ; CODE XREF: .text:00B14ADF↑p
.text:00B14C10                                         ; .text:00B14B1F↑p ...
.text:00B14C10
.text:00B14C10 var_C           = dword ptr -0Ch
.text:00B14C10 var_8           = dword ptr -8
.text:00B14C10 var_4           = dword ptr -4
.text:00B14C10 arg_0           = dword ptr  8
.text:00B14C10 arg_4           = dword ptr  0Ch
.text:00B14C10 arg_8           = dword ptr  10h
.text:00B14C10
.text:00B14C10                 push    ebp
.text:00B14C11                 mov     ebp, esp
.text:00B14C13                 sub     esp, 0Ch
.text:00B14C16                 mov     [ebp+var_8], offset unk_B1A7C0
.text:00B14C1D                 mov     eax, [ebp+var_8]
.text:00B14C20                 add     eax, [ebp+arg_4]
.text:00B14C23                 mov     [ebp+var_C], eax
.text:00B14C26                 mov     ecx, [ebp+arg_0]
.text:00B14C29                 sub     ecx, [ebp+var_C]
.text:00B14C2C                 cmp     ecx, [ebp+arg_8]
.text:00B14C2F                 jbe     short loc_B14C3A
.text:00B14C31                 mov     [ebp+var_4], 1
.text:00B14C38                 jmp     short loc_B14C41
.text:00B14C3A ; ────────────────────────────────────────────────────────────────────────────
.text:00B14C3A
.text:00B14C3A loc_B14C3A:                             ; CODE XREF: sub_B14C10+1F↑j
.text:00B14C3A                 mov     [ebp+var_4], 0
.text:00B14C41
.text:00B14C41 loc_B14C41:                             ; CODE XREF: sub_B14C10+28↑j
.text:00B14C41                 mov     eax, [ebp+var_4]
.text:00B14C44                 mov     esp, ebp
.text:00B14C46                 pop     ebp
.text:00B14C47                 retn
.text:00B14C47 sub_B14C10      endp
.text:00B14C47
.text:00B14C47 ; ────────────────────────────────────────────────────────────────────────────
.text:00B14C48                 align 20h
.text:00B14C60
.text:00B14C60 loc_B14C60:                             ; CODE XREF: .text:00B1105F↑j
.text:00B14C60                 push    ebp
.text:00B14C61                 mov     ebp, esp
.text:00B14C63                 push    ecx
.text:00B14C64                 push    offset ___guard_check_icall_fptr
.text:00B14C69                 call    sub_B110B9
.text:00B14C6E                 add     esp, 4
.text:00B14C71                 cmp     eax, offset sub_B112CB
.text:00B14C76                 jz      short loc_B14C81
.text:00B14C78                 mov     dword ptr [ebp-4], 1
```

```
.text:00B14C7F                 jmp     short loc_B14C88
.text:00B14C81 ; ───────────────────────────────────────────────────────────────────────────
.text:00B14C81
.text:00B14C81 loc_B14C81:                             ; CODE XREF: .text:00B14C76↑j
.text:00B14C81                 mov     dword ptr [ebp-4], 0
.text:00B14C88
.text:00B14C88 loc_B14C88:                             ; CODE XREF: .text:00B14C7F↑j
.text:00B14C88                 mov     eax, [ebp-4]
.text:00B14C8B                 mov     esp, ebp
.text:00B14C8D                 pop     ebp
.text:00B14C8E                 retn
.text:00B14C8E ; ───────────────────────────────────────────────────────────────────────────
.text:00B14C8F                 align 20h
.text:00B14CA0
.text:00B14CA0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B14CA0
.text:00B14CA0 ; Attributes: bp-based frame
.text:00B14CA0
.text:00B14CA0 sub_B14CA0      proc near               ; CODE XREF: sub_B15240+C9↓p
.text:00B14CA0
.text:00B14CA0 FullPath        = word ptr -414h
.text:00B14CA0 Buffer          = word ptr -20Ch
.text:00B14CA0 var_4           = dword ptr -4
.text:00B14CA0
.text:00B14CA0                 push    ebp
.text:00B14CA1                 mov     ebp, esp
.text:00B14CA3                 sub     esp, 414h
.text:00B14CA9                 mov     eax, ___security_cookie
.text:00B14CAE                 xor     eax, ebp
.text:00B14CB0                 mov     [ebp+var_4], eax
.text:00B14CB3                 cmp     byte_B1C570, 0
.text:00B14CBA                 jnz     loc_B14DDA
.text:00B14CC0                 mov     byte_B1C570, 1
.text:00B14CC7                 call    sub_B14E40
.text:00B14CCC                 test    eax, eax
.text:00B14CCE                 jnz     loc_B14DDC
.text:00B14CD4                 push    offset aVcruntime140dD_0 ; "VCRUNTIME140D.dll"
.text:00B14CD9                 call    j___vcrt_GetModuleHandleW
.text:00B14CDE                 add     esp, 4
.text:00B14CE1                 test    eax, eax
.text:00B14CE3                 jz      short loc_B14D5D
.text:00B14CE5                 push    104h
.text:00B14CEA                 lea     ecx, [ebp+FullPath]
.text:00B14CF0                 push    ecx
.text:00B14CF1                 push    eax
.text:00B14CF2                 call    j___vcrt_GetModuleFileNameW
.text:00B14CF7                 add     esp, 0Ch
.text:00B14CFA                 test    eax, eax
.text:00B14CFC                 jz      short loc_B14D5D
.text:00B14CFE                 push    104h            ; BufferCount
.text:00B14D03                 lea     eax, [ebp+Buffer]
.text:00B14D09                 push    eax             ; Buffer
.text:00B14D0A                 lea     eax, [ebp+FullPath]
.text:00B14D10                 push    eax             ; FullPath
.text:00B14D11                 call    sub_B15130
.text:00B14D16                 add     esp, 0Ch
.text:00B14D19                 test    eax, eax
.text:00B14D1B                 jz      short loc_B14D5D
.text:00B14D1D                 push    900h
.text:00B14D22                 lea     eax, [ebp+Buffer]
.text:00B14D28                 push    0
.text:00B14D2A                 push    eax
.text:00B14D2B                 call    j___vcrt_LoadLibraryExW
.text:00B14D30                 add     esp, 0Ch
.text:00B14D33                 test    eax, eax
.text:00B14D35                 jnz     loc_B14DDC
.text:00B14D3B                 call    ds:GetLastError
.text:00B14D41                 cmp     eax, 57h ; 'W'
.text:00B14D44                 jnz     short loc_B14D5D
.text:00B14D46                 push    8
.text:00B14D48                 lea     eax, [ebp+Buffer]
.text:00B14D4E                 push    0
.text:00B14D50                 push    eax
.text:00B14D51                 call    j___vcrt_LoadLibraryExW
.text:00B14D56                 add     esp, 0Ch
```

```
.text:00B14D59                 test    eax, eax
.text:00B14D5B                 jnz     short loc_B14DDC
.text:00B14D5D
.text:00B14D5D loc_B14D5D:                             ; CODE XREF: sub_B14CA0+43↑j
.text:00B14D5D                                         ; sub_B14CA0+5C↑j ...
.text:00B14D5D                 push    0A00h
.text:00B14D62                 push    0
.text:00B14D64                 push    offset aMspdb140 ; "MSPDB140"
.text:00B14D69                 call    j___vcrt_LoadLibraryExW
.text:00B14D6E                 add     esp, 0Ch
.text:00B14D71                 test    eax, eax
.text:00B14D73                 jnz     short loc_B14DDC
.text:00B14D75                 call    ds:GetLastError
.text:00B14D7B                 cmp     eax, 57h ; 'W'
.text:00B14D7E                 jnz     short loc_B14DDA
.text:00B14D80                 push    104h
.text:00B14D85                 lea     eax, [ebp+FullPath]
.text:00B14D8B                 push    eax
.text:00B14D8C                 push    0
.text:00B14D8E                 call    j___vcrt_GetModuleFileNameW
.text:00B14D93                 add     esp, 0Ch
.text:00B14D96                 test    eax, eax
.text:00B14D98                 jz      short loc_B14DDA
.text:00B14D9A                 push    104h            ; BufferCount
.text:00B14D9F                 lea     eax, [ebp+Buffer]
.text:00B14DA5                 push    eax             ; Buffer
.text:00B14DA6                 lea     eax, [ebp+FullPath]
.text:00B14DAC                 push    eax             ; FullPath
.text:00B14DAD                 call    sub_B15130
.text:00B14DB2                 add     esp, 0Ch
.text:00B14DB5                 test    eax, eax
.text:00B14DB7                 jz      short loc_B14DDA
.text:00B14DB9                 push    8
.text:00B14DBB                 lea     eax, [ebp+Buffer]
.text:00B14DC1                 push    0
.text:00B14DC3                 push    eax
.text:00B14DC4                 call    j___vcrt_LoadLibraryExW
.text:00B14DC9                 add     esp, 0Ch
.text:00B14DCC                 mov     ecx, [ebp+var_4]
.text:00B14DCF                 xor     ecx, ebp        ; StackCookie
.text:00B14DD1                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B14DD6                 mov     esp, ebp
.text:00B14DD8                 pop     ebp
.text:00B14DD9                 retn
.text:00B14DDA ; ---------------------------------------------------------------------------
.text:00B14DDA
.text:00B14DDA loc_B14DDA:                             ; CODE XREF: sub_B14CA0+1A↑j
.text:00B14DDA                                         ; sub_B14CA0+DE↑j ...
.text:00B14DDA                 xor     eax, eax
.text:00B14DDC
.text:00B14DDC loc_B14DDC:                             ; CODE XREF: sub_B14CA0+2E↑j
.text:00B14DDC                                         ; sub_B14CA0+95↑j ...
.text:00B14DDC                 mov     ecx, [ebp+var_4]
.text:00B14DDF                 xor     ecx, ebp        ; StackCookie
.text:00B14DE1                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B14DE6                 mov     esp, ebp
.text:00B14DE8                 pop     ebp
.text:00B14DE9                 retn
.text:00B14DE9 sub_B14CA0      endp
.text:00B14DE9
.text:00B14DE9 ; ---------------------------------------------------------------------------
.text:00B14DEA                 db 56h dup(0CCh)
.text:00B14E40
.text:00B14E40 ; =============== S U B R O U T I N E =======================================
.text:00B14E40
.text:00B14E40 ; Attributes: bp-based frame
.text:00B14E40
.text:00B14E40 sub_B14E40      proc near               ; CODE XREF: sub_B14CA0+27↑p
.text:00B14E40
.text:00B14E40 var_21C         = dword ptr -21Ch
.text:00B14E40 var_218         = dword ptr -218h
.text:00B14E40 var_214         = dword ptr -214h
.text:00B14E40 var_210         = dword ptr -210h
.text:00B14E40 var_20C         = dword ptr -20Ch
.text:00B14E40 var_208         = dword ptr -208h
```

```
.text:00B14E40 var_204         = dword ptr -204h
.text:00B14E40 var_200         = dword ptr -200h
.text:00B14E40 var_1FC         = dword ptr -1FCh
.text:00B14E40 var_1F8         = dword ptr -1F8h
.text:00B14E40 var_1F4         = dword ptr -1F4h
.text:00B14E40 var_1F0         = dword ptr -1F0h
.text:00B14E40 var_1EC         = word ptr -1ECh
.text:00B14E40 var_4           = dword ptr -4
.text:00B14E40
.text:00B14E40                 push    ebp
.text:00B14E41                 mov     ebp, esp
.text:00B14E43                 sub     esp, 21Ch
.text:00B14E49                 mov     eax, ___security_cookie
.text:00B14E4E                 xor     eax, ebp
.text:00B14E50                 mov     [ebp+var_4], eax
.text:00B14E53                 push    ebx
.text:00B14E54                 push    esi
.text:00B14E55                 push    edi
.text:00B14E56                 push    800h
.text:00B14E5B                 push    0
.text:00B14E5D                 push    offset aApiMsWinCoreRe ; "api-ms-win-core-registry-l1-1-0.dll"
.text:00B14E62                 call    j___vcrt_LoadLibraryExW
.text:00B14E67                 mov     edi, eax
.text:00B14E69                 add     esp, 0Ch
.text:00B14E6C                 test    edi, edi
.text:00B14E6E                 jnz     short loc_B14EA9
.text:00B14E70                 push    800h
.text:00B14E75                 push    eax
.text:00B14E76                 push    offset aAdvapi32Dll ; "advapi32.dll"
.text:00B14E7B                 call    j___vcrt_LoadLibraryExW
.text:00B14E80                 mov     edi, eax
.text:00B14E82                 add     esp, 0Ch
.text:00B14E85                 test    edi, edi
.text:00B14E87                 jnz     short loc_B14EA9
.text:00B14E89                 call    ds:GetLastError
.text:00B14E8F                 cmp     eax, 57h ; 'W'
.text:00B14E92                 jnz     short loc_B14F0D
.text:00B14E94                 push    edi
.text:00B14E95                 push    edi
.text:00B14E96                 push    offset aAdvapi32Dll ; "advapi32.dll"
.text:00B14E9B                 call    j___vcrt_LoadLibraryExW
.text:00B14EA0                 mov     edi, eax
.text:00B14EA2                 add     esp, 0Ch
.text:00B14EA5                 test    edi, edi
.text:00B14EA7                 jz      short loc_B14F0D
.text:00B14EA9
.text:00B14EA9 loc_B14EA9:                             ; CODE XREF: sub_B14E40+2E↑j
.text:00B14EA9                                         ; sub_B14E40+47↑j
.text:00B14EA9                 push    offset ProcName ; "RegOpenKeyExW"
.text:00B14EAE                 push    edi             ; hModule
.text:00B14EAF                 call    ds:GetProcAddress
.text:00B14EB5                 mov     esi, eax
.text:00B14EB7                 test    esi, esi
.text:00B14EB9                 jz      short loc_B14F0D
.text:00B14EBB                 push    offset aRegqueryvaluee ; "RegQueryValueExW"
.text:00B14EC0                 push    edi             ; hModule
.text:00B14EC1                 call    ds:GetProcAddress
.text:00B14EC7                 mov     ebx, eax
.text:00B14EC9                 test    ebx, ebx
.text:00B14ECB                 jz      short loc_B14F0D
.text:00B14ECD                 push    offset aRegclosekey ; "RegCloseKey"
.text:00B14ED2                 push    edi             ; hModule
.text:00B14ED3                 call    ds:GetProcAddress
.text:00B14ED9                 mov     [ebp+var_218], eax
.text:00B14EDF                 test    eax, eax
.text:00B14EE1                 jz      short loc_B14F0D
.text:00B14EE3                 lea     eax, [ebp+var_210]
.text:00B14EE9                 mov     ecx, esi
.text:00B14EEB                 push    eax
.text:00B14EEC                 push    1
.text:00B14EEE                 push    0
.text:00B14EF0                 push    offset aSoftwareWow643 ; "SOFTWARE\\Wow6432Node\\Microsoft\\Visua"...
.text:00B14EF5                 push    80000002h
.text:00B14EFA                 call    ds:___guard_check_icall_fptr
.text:00B14F00                 call    esi
```

```
.text:00B14F02                 test    eax, eax
.text:00B14F04                 jz      short loc_B14F20
.text:00B14F06                 push    edi             ; hLibModule
.text:00B14F07                 call    ds:FreeLibrary
.text:00B14F0D
.text:00B14F0D loc_B14F0D:                             ; CODE XREF: sub_B14E40+52↑j
.text:00B14F0D                                         ; sub_B14E40+67↑j ...
.text:00B14F0D                 pop     edi
.text:00B14F0E                 pop     esi
.text:00B14F0F                 xor     eax, eax
.text:00B14F11                 pop     ebx
.text:00B14F12                 mov     ecx, [ebp+var_4]
.text:00B14F15                 xor     ecx, ebp        ; StackCookie
.text:00B14F17                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B14F1C                 mov     esp, ebp
.text:00B14F1E                 pop     ebp
.text:00B14F1F                 retn
.text:00B14F20 ; ---------------------------------------------------------------------------
.text:00B14F20
.text:00B14F20 loc_B14F20:                             ; CODE XREF: sub_B14E40+C4↑j
.text:00B14F20                 lea     eax, [ebp+var_214]
.text:00B14F26                 mov     [ebp+var_214], 208h
.text:00B14F30                 push    eax
.text:00B14F31                 lea     eax, [ebp+var_20C]
.text:00B14F37                 mov     ecx, ebx
.text:00B14F39                 push    eax
.text:00B14F3A                 lea     eax, [ebp+var_21C]
.text:00B14F40                 push    eax
.text:00B14F41                 push    0
.text:00B14F43                 push    offset aProductdir ; "ProductDir"
.text:00B14F48                 push    [ebp+var_210]
.text:00B14F4E                 call    ds:___guard_check_icall_fptr
.text:00B14F54                 call    ebx
.text:00B14F56                 push    [ebp+var_210]
.text:00B14F5C                 mov     ebx, [ebp+var_218]
.text:00B14F62                 mov     esi, eax
.text:00B14F64                 mov     ecx, ebx
.text:00B14F66                 call    ds:___guard_check_icall_fptr
.text:00B14F6C                 call    ebx
.text:00B14F6E                 push    edi             ; hLibModule
.text:00B14F6F                 call    ds:FreeLibrary
.text:00B14F75                 test    esi, esi
.text:00B14F77                 jnz     short loc_B14F0D
.text:00B14F79                 cmp     [ebp+var_21C], 1
.text:00B14F80                 jnz     short loc_B14F0D
.text:00B14F82                 mov     ecx, [ebp+var_214]
.text:00B14F88                 test    cl, 1
.text:00B14F8B                 jnz     short loc_B14F0D
.text:00B14F8D                 shr     ecx, 1
.text:00B14F8F                 mov     edx, ecx
.text:00B14F91                 cmp     ecx, 2
.text:00B14F94                 jb      loc_B14F0D
.text:00B14F9A                 dec     ecx
.text:00B14F9B                 lea     eax, [ebp+var_20C]
.text:00B14FA1                 cmp     [eax+ecx*2], si
.text:00B14FA5                 lea     eax, [eax+ecx*2]
.text:00B14FA8                 jnz     loc_B14F0D
.text:00B14FAE                 cmp     word ptr [eax-2], 5Ch ; '\'
.text:00B14FB3                 jz      short loc_B14FBF
.text:00B14FB5                 mov     ecx, 5Ch ; '\'
.text:00B14FBA                 mov     [eax], cx
.text:00B14FBD                 mov     ecx, edx
.text:00B14FBF
.text:00B14FBF loc_B14FBF:                             ; CODE XREF: sub_B14E40+173↑j
.text:00B14FBF                 mov     eax, ecx
.text:00B14FC1                 not     eax
.text:00B14FC3                 cmp     eax, 12h
.text:00B14FC6                 jb      loc_B14F0D
.text:00B14FCC                 lea     eax, [ecx+11h]
.text:00B14FCF                 cmp     eax, 104h
.text:00B14FD4                 ja      loc_B14F0D
.text:00B14FDA                 mov     eax, ds:dword_B1A2F0
.text:00B14FDF                 mov     [ebp+ecx*2+var_20C], eax
.text:00B14FE6                 mov     eax, ds:dword_B1A2F4
.text:00B14FEB                 mov     [ebp+ecx*2+var_208], eax
```

```
.text:00B14FF2                 mov     eax, ds:dword_B1A2F8
.text:00B14FF7                 mov     [ebp+ecx*2+var_204], eax
.text:00B14FFE                 mov     eax, ds:dword_B1A2FC
.text:00B15003                 mov     [ebp+ecx*2+var_200], eax
.text:00B1500A                 mov     eax, ds:dword_B1A300
.text:00B1500F                 mov     [ebp+ecx*2+var_1FC], eax
.text:00B15016                 mov     eax, ds:dword_B1A304
.text:00B1501B                 mov     [ebp+ecx*2+var_1F8], eax
.text:00B15022                 mov     eax, ds:dword_B1A308
.text:00B15027                 mov     [ebp+ecx*2+var_1F4], eax
.text:00B1502E                 mov     eax, ds:dword_B1A30C
.text:00B15033                 mov     [ebp+ecx*2+var_1F0], eax
.text:00B1503A                 mov     ax, ds:word_B1A310
.text:00B15040                 push    900h
.text:00B15045                 mov     [ebp+ecx*2+var_1EC], ax
.text:00B1504D                 lea     eax, [ebp+var_20C]
.text:00B15053                 push    0
.text:00B15055                 push    eax
.text:00B15056                 call    j___vcrt_LoadLibraryExW
.text:00B1505B                 mov     esi, eax
.text:00B1505D                 add     esp, 0Ch
.text:00B15060                 test    esi, esi
.text:00B15062                 jnz     short loc_B15083
.text:00B15064                 call    ds:GetLastError
.text:00B1506A                 cmp     eax, 57h ; 'W'
.text:00B1506D                 jnz     short loc_B15083
.text:00B1506F                 push    8
.text:00B15071                 lea     eax, [ebp+var_20C]
.text:00B15077                 push    esi
.text:00B15078                 push    eax
.text:00B15079                 call    j___vcrt_LoadLibraryExW
.text:00B1507E                 add     esp, 0Ch
.text:00B15081                 mov     esi, eax
.text:00B15083
.text:00B15083 loc_B15083:                             ; CODE XREF: sub_B14E40+222↑j
.text:00B15083                                         ; sub_B14E40+22D↑j
.text:00B15083                 mov     ecx, [ebp+var_4]
.text:00B15086                 mov     eax, esi
.text:00B15088                 pop     edi
.text:00B15089                 pop     esi
.text:00B1508A                 xor     ecx, ebp        ; StackCookie
.text:00B1508C                 pop     ebx
.text:00B1508D                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B15092                 mov     esp, ebp
.text:00B15094                 pop     ebp
.text:00B15095                 retn
.text:00B15095 sub_B14E40      endp
.text:00B15095
.text:00B15095 ; ---------------------------------------------------------------------------
.text:00B15096                 db 9Ah dup(0CCh)
.text:00B15130
.text:00B15130 ; =============== S U B R O U T I N E =======================================
.text:00B15130
.text:00B15130 ; Attributes: bp-based frame
.text:00B15130
.text:00B15130 ; int __cdecl sub_B15130(wchar_t *FullPath, wchar_t *Buffer, size_t BufferCount)
.text:00B15130 sub_B15130      proc near               ; CODE XREF: sub_B14CA0+71↑p
.text:00B15130                                         ; sub_B14CA0+10D↑p
.text:00B15130
.text:00B15130 Dir             = word ptr -60Ch
.text:00B15130 Filename        = word ptr -40Ch
.text:00B15130 Ext             = word ptr -20Ch
.text:00B15130 Drive           = word ptr -0Ch
.text:00B15130 var_4           = dword ptr -4
.text:00B15130 FullPath        = dword ptr  8
.text:00B15130 Buffer          = dword ptr  0Ch
.text:00B15130 BufferCount     = dword ptr  10h
.text:00B15130
.text:00B15130                 push    ebp
.text:00B15131                 mov     ebp, esp
.text:00B15133                 sub     esp, 60Ch
.text:00B15139                 mov     eax, ___security_cookie
.text:00B1513E                 xor     eax, ebp
.text:00B15140                 mov     [ebp+var_4], eax
.text:00B15143                 mov     eax, [ebp+FullPath]
```

```
.text:00B15146                 xor     ecx, ecx
.text:00B15148                 push    esi
.text:00B15149                 mov     esi, [ebp+Buffer]
.text:00B1514C                 push    100h            ; ExtCount
.text:00B15151                 mov     [esi], cx
.text:00B15154                 lea     ecx, [ebp+Ext]
.text:00B1515A                 push    ecx             ; Ext
.text:00B1515B                 push    100h            ; FilenameCount
.text:00B15160                 lea     ecx, [ebp+Filename]
.text:00B15166                 push    ecx             ; Filename
.text:00B15167                 push    100h            ; DirCount
.text:00B1516C                 lea     ecx, [ebp+Dir]
.text:00B15172                 push    ecx             ; Dir
.text:00B15173                 push    3               ; DriveCount
.text:00B15175                 lea     ecx, [ebp+Drive]
.text:00B15178                 push    ecx             ; Drive
.text:00B15179                 push    eax             ; FullPath
.text:00B1517A                 call    j__wsplitpath_s
.text:00B1517F                 add     esp, 24h
.text:00B15182                 test    eax, eax
.text:00B15184                 jnz     short loc_B151F3
.text:00B15186                 push    offset aMspdb140_0 ; "MSPDB140"
.text:00B1518B                 lea     eax, [ebp+Filename]
.text:00B15191                 push    9               ; SizeInWords
.text:00B15193                 push    eax             ; Destination
.text:00B15194                 call    j_wcscpy_s
.text:00B15199                 add     esp, 0Ch
.text:00B1519C                 test    eax, eax
.text:00B1519E                 jnz     short loc_B151F3
.text:00B151A0                 push    offset aDll     ; "DLL"
.text:00B151A5                 lea     eax, [ebp+Ext]
.text:00B151AB                 push    4               ; SizeInWords
.text:00B151AD                 push    eax             ; Destination
.text:00B151AE                 call    j_wcscpy_s
.text:00B151B3                 add     esp, 0Ch
.text:00B151B6                 test    eax, eax
.text:00B151B8                 jnz     short loc_B151F3
.text:00B151BA                 lea     eax, [ebp+Ext]
.text:00B151C0                 push    eax             ; Ext
.text:00B151C1                 lea     eax, [ebp+Filename]
.text:00B151C7                 push    eax             ; Filename
.text:00B151C8                 lea     eax, [ebp+Dir]
.text:00B151CE                 push    eax             ; Dir
.text:00B151CF                 lea     eax, [ebp+Drive]
.text:00B151D2                 push    eax             ; Drive
.text:00B151D3                 push    [ebp+BufferCount] ; BufferCount
.text:00B151D6                 push    esi             ; Buffer
.text:00B151D7                 call    j__wmakepath_s
.text:00B151DC                 add     esp, 18h
.text:00B151DF                 neg     eax
.text:00B151E1                 sbb     eax, eax
.text:00B151E3                 inc     eax
.text:00B151E4                 pop     esi
.text:00B151E5                 mov     ecx, [ebp+var_4]
.text:00B151E8                 xor     ecx, ebp        ; StackCookie
.text:00B151EA                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B151EF                 mov     esp, ebp
.text:00B151F1                 pop     ebp
.text:00B151F2                 retn
.text:00B151F3 ; ───────────────────────────────────────────────────────────────────────────
.text:00B151F3
.text:00B151F3 loc_B151F3:                             ; CODE XREF: sub_B15130+54↑j
.text:00B151F3                                         ; sub_B15130+6E↑j ...
.text:00B151F3                 mov     ecx, [ebp+var_4]
.text:00B151F6                 xor     eax, eax
.text:00B151F8                 xor     ecx, ebp        ; StackCookie
.text:00B151FA                 pop     esi
.text:00B151FB                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B15200                 mov     esp, ebp
.text:00B15202                 pop     ebp
.text:00B15203                 retn
.text:00B15203 sub_B15130      endp
.text:00B15203
.text:00B15203 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15204                 align 40h
```

```
.text:00B15240
.text:00B15240 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B15240
.text:00B15240 ; Attributes: bp-based frame
.text:00B15240
.text:00B15240 sub_B15240      proc near               ; CODE XREF: sub_B11096↑j
.text:00B15240
.text:00B15240 Buffer          = _MEMORY_BASIC_INFORMATION ptr -4Ch
.text:00B15240 var_30          = byte ptr -30h
.text:00B15240 var_2C          = dword ptr -2Ch
.text:00B15240 var_28          = dword ptr -28h
.text:00B15240 var_24          = dword ptr -24h
.text:00B15240 var_20          = dword ptr -20h
.text:00B15240 var_1C          = dword ptr -1Ch
.text:00B15240 var_18          = dword ptr -18h
.text:00B15240 var_14          = dword ptr -14h
.text:00B15240 var_10          = dword ptr -10h
.text:00B15240 var_C           = dword ptr -0Ch
.text:00B15240 var_8           = dword ptr -8
.text:00B15240 var_4           = word ptr -4
.text:00B15240 arg_0           = dword ptr  8
.text:00B15240 arg_4           = dword ptr  0Ch
.text:00B15240 arg_8           = byte ptr  10h
.text:00B15240 arg_C           = dword ptr  14h
.text:00B15240 arg_10          = dword ptr  18h
.text:00B15240 arg_14          = dword ptr  1Ch
.text:00B15240
.text:00B15240                 push    ebp
.text:00B15241                 mov     ebp, esp
.text:00B15243                 mov     eax, [ebp+arg_C]
.text:00B15246                 sub     esp, 4Ch
.text:00B15249                 xor     ecx, ecx
.text:00B1524B                 push    ebx
.text:00B1524C                 push    esi
.text:00B1524D                 mov     esi, [ebp+arg_0]
.text:00B15250                 mov     dword ptr [eax], 0
.text:00B15256                 dec     esi
.text:00B15257                 mov     eax, [ebp+arg_4]
.text:00B1525A                 push    edi
.text:00B1525B                 push    1Ch             ; dwLength
.text:00B1525D                 mov     [eax], cx
.text:00B15260                 lea     eax, [ebp+Buffer]
.text:00B15263                 push    eax             ; lpBuffer
.text:00B15264                 push    esi             ; lpAddress
.text:00B15265                 call    ds:VirtualQuery
.text:00B1526B                 test    eax, eax
.text:00B1526D                 jz      loc_B155B7
.text:00B15273                 push    [ebp+arg_14]
.text:00B15276                 push    [ebp+arg_10]
.text:00B15279                 push    [ebp+Buffer.AllocationBase]
.text:00B1527C                 call    j___vcrt_GetModuleFileNameW
.text:00B15281                 add     esp, 0Ch
.text:00B15284                 test    eax, eax
.text:00B15286                 jz      loc_B155B7
.text:00B1528C                 mov     eax, [ebp+Buffer.AllocationBase]
.text:00B1528F                 mov     ecx, 5A4Dh
.text:00B15294                 cmp     [eax], cx
.text:00B15297                 jnz     loc_B155B7
.text:00B1529D                 mov     ecx, [eax+3Ch]
.text:00B152A0                 test    ecx, ecx
.text:00B152A2                 jle     loc_B155B7
.text:00B152A8                 add     ecx, eax
.text:00B152AA                 cmp     dword ptr [ecx], 4550h
.text:00B152B0                 jnz     loc_B155B7
.text:00B152B6                 movzx   edx, word ptr [ecx+14h]
.text:00B152BA                 sub     esi, eax
.text:00B152BC                 movzx   edi, word ptr [ecx+6]
.text:00B152C0                 add     edx, 20h ; ' '
.text:00B152C3                 xor     ebx, ebx
.text:00B152C5                 xor     eax, eax
.text:00B152C7                 test    edi, edi
.text:00B152C9                 jz      short loc_B152E7
.text:00B152CB                 add     ecx, edx
.text:00B152CD                 lea     ecx, [ecx+0]
.text:00B152D0
```

```
.text:00B152D0 loc_B152D0:                            ; CODE XREF: sub_B15240+A5↓j
.text:00B152D0                 mov     edx, [ecx+4]
.text:00B152D3                 cmp     esi, edx
.text:00B152D5                 jb      short loc_B152DF
.text:00B152D7                 mov     ebx, esi
.text:00B152D9                 sub     ebx, edx
.text:00B152DB                 cmp     esi, [ecx]
.text:00B152DD                 jb      short loc_B152E7
.text:00B152DF
.text:00B152DF loc_B152DF:                            ; CODE XREF: sub_B15240+95↑j
.text:00B152DF                 inc     eax
.text:00B152E0                 add     ecx, 28h ; '('
.text:00B152E3                 cmp     eax, edi
.text:00B152E5                 jb      short loc_B152D0
.text:00B152E7
.text:00B152E7 loc_B152E7:                            ; CODE XREF: sub_B15240+89↑j
.text:00B152E7                                        ; sub_B15240+9D↑j
.text:00B152E7                 cmp     eax, edi
.text:00B152E9                 jz      loc_B155B7
.text:00B152EF                 inc     eax
.text:00B152F0                 cmp     byte_B1C571, 0
.text:00B152F7                 mov     [ebp+var_C], eax
.text:00B152FA                 jnz     short loc_B15324
.text:00B152FC                 cmp     dword_B1C56C, 0
.text:00B15303                 jnz     loc_B155B7
.text:00B15309                 call    sub_B14CA0
.text:00B1530E                 mov     dword_B1C56C, eax
.text:00B15313                 test    eax, eax
.text:00B15315                 jz      loc_B155B7
.text:00B1531B                 mov     byte_B1C571, 1
.text:00B15322                 jmp     short loc_B15329
.text:00B15324 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15324
.text:00B15324 loc_B15324:                            ; CODE XREF: sub_B15240+BA↑j
.text:00B15324                 mov     eax, dword_B1C56C
.text:00B15329
.text:00B15329 loc_B15329:                            ; CODE XREF: sub_B15240+E2↑j
.text:00B15329                 push    offset aPdbopenvalidat ; "PDBOpenValidate5"
.text:00B1532E                 push    eax             ; hModule
.text:00B1532F                 call    ds:GetProcAddress
.text:00B15335                 mov     esi, eax
.text:00B15337                 test    esi, esi
.text:00B15339                 jz      loc_B155B7
.text:00B1533F                 lea     eax, [ebp+var_18]
.text:00B15342                 mov     ecx, esi
.text:00B15344                 push    eax
.text:00B15345                 push    0
.text:00B15347                 push    0
.text:00B15349                 lea     eax, [ebp+var_30]
.text:00B1534C                 push    eax
.text:00B1534D                 push    0
.text:00B1534F                 push    0
.text:00B15351                 push    0
.text:00B15353                 push    [ebp+arg_10]
.text:00B15356                 call    ds:___guard_check_icall_fptr
.text:00B1535C                 call    esi
.text:00B1535E                 add     esp, 20h
.text:00B15361                 test    eax, eax
.text:00B15363                 jz      loc_B155B7
.text:00B15369                 mov     edi, [ebp+var_18]
.text:00B1536C                 mov     [ebp+var_8], 0
.text:00B15373                 mov     [ebp+var_24], 0
.text:00B1537A                 mov     eax, [edi]
.text:00B1537C                 mov     esi, [eax]
.text:00B1537E                 mov     ecx, esi
.text:00B15380                 call    ds:___guard_check_icall_fptr
.text:00B15386                 mov     ecx, edi
.text:00B15388                 call    esi
.text:00B1538A                 cmp     eax, 1329141h
.text:00B1538F                 jnz     loc_B154D4
.text:00B15395                 mov     edi, [ebp+var_18]
.text:00B15398                 mov     eax, [edi]
.text:00B1539A                 mov     esi, [eax+1Ch]
.text:00B1539D                 lea     eax, [ebp+var_20]
.text:00B153A0                 push    eax
```

```
.text:00B153A1                 push    offset aR       ; "r"
.text:00B153A6                 push    0
.text:00B153A8                 mov     ecx, esi
.text:00B153AA                 call    ds:___guard_check_icall_fptr
.text:00B153B0                 mov     ecx, edi
.text:00B153B2                 call    esi
.text:00B153B4                 test    eax, eax
.text:00B153B6                 jz      loc_B154D4
.text:00B153BC                 mov     edi, [ebp+var_20]
.text:00B153BF                 push    0
.text:00B153C1                 push    0
.text:00B153C3                 push    0
.text:00B153C5                 mov     eax, [edi]
.text:00B153C7                 mov     esi, [eax+20h]
.text:00B153CA                 lea     eax, [ebp+var_14]
.text:00B153CD                 push    eax
.text:00B153CE                 push    ebx
.text:00B153CF                 push    [ebp+var_C]
.text:00B153D2                 mov     ecx, esi
.text:00B153D4                 call    ds:___guard_check_icall_fptr
.text:00B153DA                 mov     ecx, edi
.text:00B153DC                 call    esi
.text:00B153DE                 test    eax, eax
.text:00B153E0                 jz      loc_B154C0
.text:00B153E6                 mov     edi, [ebp+var_14]
.text:00B153E9                 mov     eax, [edi]
.text:00B153EB                 mov     esi, [eax+68h]
.text:00B153EE                 lea     eax, [ebp+var_8]
.text:00B153F1                 push    eax
.text:00B153F2                 mov     ecx, esi
.text:00B153F4                 call    ds:___guard_check_icall_fptr
.text:00B153FA                 mov     ecx, edi
.text:00B153FC                 call    esi
.text:00B153FE                 test    al, al
.text:00B15400                 jz      loc_B154AC
.text:00B15406                 mov     edi, [ebp+var_8]
.text:00B15409                 test    edi, edi
.text:00B1540B                 jz      loc_B154AC
.text:00B15411                 mov     eax, [edi]
.text:00B15413                 mov     esi, [eax+8]
.text:00B15416                 mov     ecx, esi
.text:00B15418                 call    ds:___guard_check_icall_fptr
.text:00B1541E                 mov     ecx, edi
.text:00B15420                 call    esi
.text:00B15422                 test    eax, eax
.text:00B15424                 jz      short loc_B15487
.text:00B15426
.text:00B15426 loc_B15426:                             ; CODE XREF: sub_B15240+245↓j
.text:00B15426                 mov     edi, [ebp+var_8]
.text:00B15429                 push    0
.text:00B1542B                 mov     eax, [edi]
.text:00B1542D                 mov     esi, [eax+0Ch]
.text:00B15430                 lea     eax, [ebp+var_10]
.text:00B15433                 push    eax
.text:00B15434                 lea     eax, [ebp+var_28]
.text:00B15437                 mov     ecx, esi
.text:00B15439                 push    eax
.text:00B1543A                 lea     eax, [ebp+var_4]
.text:00B1543D                 push    eax
.text:00B1543E                 lea     eax, [ebp+var_1C]
.text:00B15441                 push    eax
.text:00B15442                 push    0
.text:00B15444                 call    ds:___guard_check_icall_fptr
.text:00B1544A                 mov     ecx, edi
.text:00B1544C                 call    esi
.text:00B1544E                 test    al, al
.text:00B15450                 jz      short loc_B15499
.text:00B15452                 movzx   eax, [ebp+var_4]
.text:00B15456                 cmp     eax, [ebp+var_C]
.text:00B15459                 jnz     short loc_B1546F
.text:00B1545B                 mov     ecx, [ebp+var_1C]
.text:00B1545E                 cmp     ecx, ebx
.text:00B15460                 ja      short loc_B1546F
.text:00B15462                 mov     eax, [ebp+var_28]
.text:00B15465                 add     eax, ecx
```

```
.text:00B15467                 cmp     ebx, eax
.text:00B15469                 jb      loc_B154F2
.text:00B1546F
.text:00B1546F loc_B1546F:                             ; CODE XREF: sub_B15240+219↑j
.text:00B1546F                                         ; sub_B15240+220↑j
.text:00B1546F                 mov     edi, [ebp+var_8]
.text:00B15472                 mov     eax, [edi]
.text:00B15474                 mov     esi, [eax+8]
.text:00B15477                 mov     ecx, esi
.text:00B15479                 call    ds:___guard_check_icall_fptr
.text:00B1547F                 mov     ecx, edi
.text:00B15481                 call    esi
.text:00B15483                 test    eax, eax
.text:00B15485                 jnz     short loc_B15426
.text:00B15487
.text:00B15487 loc_B15487:                             ; CODE XREF: sub_B15240+1E4↑j
.text:00B15487                 xor     ecx, ecx
.text:00B15489
.text:00B15489 loc_B15489:                             ; CODE XREF: sub_B15240+307↓j
.text:00B15489                                         ; sub_B15240+312↓j ...
.text:00B15489                 push    ecx             ; lpMem
.text:00B1548A                 push    0               ; dwFlags
.text:00B1548C                 call    ds:GetProcessHeap
.text:00B15492                 push    eax             ; hHeap
.text:00B15493                 call    ds:HeapFree
.text:00B15499
.text:00B15499 loc_B15499:                             ; CODE XREF: sub_B15240+210↑j
.text:00B15499                                         ; sub_B15240+2B7↓j ...
.text:00B15499                 mov     edi, [ebp+var_8]
.text:00B1549C                 mov     eax, [edi]
.text:00B1549E                 mov     esi, [eax]
.text:00B154A0                 mov     ecx, esi
.text:00B154A2                 call    ds:___guard_check_icall_fptr
.text:00B154A8                 mov     ecx, edi
.text:00B154AA                 call    esi
.text:00B154AC
.text:00B154AC loc_B154AC:                             ; CODE XREF: sub_B15240+1C0↑j
.text:00B154AC                                         ; sub_B15240+1CB↑j
.text:00B154AC                 mov     edi, [ebp+var_14]
.text:00B154AF                 mov     eax, [edi]
.text:00B154B1                 mov     esi, [eax+40h]
.text:00B154B4                 mov     ecx, esi
.text:00B154B6                 call    ds:___guard_check_icall_fptr
.text:00B154BC                 mov     ecx, edi
.text:00B154BE                 call    esi
.text:00B154C0
.text:00B154C0 loc_B154C0:                             ; CODE XREF: sub_B15240+1A0↑j
.text:00B154C0                 mov     edi, [ebp+var_20]
.text:00B154C3                 mov     eax, [edi]
.text:00B154C5                 mov     esi, [eax+38h]
.text:00B154C8                 mov     ecx, esi
.text:00B154CA                 call    ds:___guard_check_icall_fptr
.text:00B154D0                 mov     ecx, edi
.text:00B154D2                 call    esi
.text:00B154D4
.text:00B154D4 loc_B154D4:                             ; CODE XREF: sub_B15240+14F↑j
.text:00B154D4                                         ; sub_B15240+176↑j
.text:00B154D4                 mov     edi, [ebp+var_18]
.text:00B154D7                 mov     edx, [edi]
.text:00B154D9                 mov     esi, [edx+2Ch]
.text:00B154DC                 mov     ecx, esi
.text:00B154DE                 call    ds:___guard_check_icall_fptr
.text:00B154E4                 mov     ecx, edi
.text:00B154E6                 call    esi
.text:00B154E8                 mov     eax, [ebp+var_24]
.text:00B154EB                 pop     edi
.text:00B154EC                 pop     esi
.text:00B154ED                 pop     ebx
.text:00B154EE                 mov     esp, ebp
.text:00B154F0                 pop     ebp
.text:00B154F1                 retn
.text:00B154F2 ; ───────────────────────────────────────────────────────────────────────────
.text:00B154F2
.text:00B154F2 loc_B154F2:                             ; CODE XREF: sub_B15240+229↑j
.text:00B154F2                 mov     esi, [ebp+var_10]
```

```
.text:00B154F5                 test    esi, esi
.text:00B154F7                 jz      short loc_B15499
.text:00B154F9                 cmp     esi, 1FFFFFFFh
.text:00B154FF                 jnb     short loc_B15499
.text:00B15501                 shl     esi, 3
.text:00B15504                 call    ds:GetProcessHeap
.text:00B1550A                 push    esi             ; dwBytes
.text:00B1550B                 push    0               ; dwFlags
.text:00B1550D                 push    eax             ; hHeap
.text:00B1550E                 call    ds:HeapAlloc
.text:00B15514                 mov     [ebp+var_C], eax
.text:00B15517                 test    eax, eax
.text:00B15519                 jz      loc_B15499
.text:00B1551F                 mov     edi, [ebp+var_8]
.text:00B15522                 push    eax
.text:00B15523                 lea     eax, [ebp+var_10]
.text:00B15526                 push    eax
.text:00B15527                 mov     ecx, [edi]
.text:00B15529                 lea     eax, [ebp+var_2C]
.text:00B1552C                 push    0
.text:00B1552E                 push    0
.text:00B15530                 push    0
.text:00B15532                 mov     esi, [ecx+0Ch]
.text:00B15535                 mov     ecx, esi
.text:00B15537                 push    eax
.text:00B15538                 call    ds:___guard_check_icall_fptr
.text:00B1553E                 mov     ecx, edi
.text:00B15540                 call    esi
.text:00B15542                 mov     ecx, [ebp+var_C]
.text:00B15545                 test    al, al
.text:00B15547                 jz      loc_B15489
.text:00B1554D                 sub     ebx, [ebp+var_1C]
.text:00B15550                 cmp     ebx, [ecx]
.text:00B15552                 jb      loc_B15489
.text:00B15558                 mov     edx, [ebp+var_10]
.text:00B1555B                 mov     eax, 1
.text:00B15560                 cmp     edx, eax
.text:00B15562                 jbe     short loc_B1556E
.text:00B15564
.text:00B15564 loc_B15564:                             ; CODE XREF: sub_B15240+32C↓j
.text:00B15564                 cmp     ebx, [ecx+eax*8]
.text:00B15567                 jb      short loc_B1556E
.text:00B15569                 inc     eax
.text:00B1556A                 cmp     eax, edx
.text:00B1556C                 jb      short loc_B15564
.text:00B1556E
.text:00B1556E loc_B1556E:                             ; CODE XREF: sub_B15240+322↑j
.text:00B1556E                                         ; sub_B15240+327↑j
.text:00B1556E                 mov     eax, [ecx+eax*8-4]
.text:00B15572                 mov     ecx, [ebp+arg_C]
.text:00B15575                 and     eax, 0FFFFFFh
.text:00B1557A                 mov     edi, [ebp+var_14]
.text:00B1557D                 push    0
.text:00B1557F                 push    0
.text:00B15581                 mov     [ecx], eax
.text:00B15583                 mov     eax, [edi]
.text:00B15585                 push    0
.text:00B15587                 mov     esi, [eax+70h]
.text:00B1558A                 lea     eax, [ebp+arg_8]
.text:00B1558D                 push    eax
.text:00B1558E                 push    [ebp+arg_4]
.text:00B15591                 mov     ecx, esi
.text:00B15593                 push    [ebp+var_2C]
.text:00B15596                 call    ds:___guard_check_icall_fptr
.text:00B1559C                 mov     ecx, edi
.text:00B1559E                 call    esi
.text:00B155A0                 mov     ecx, [ebp+var_C]
.text:00B155A3                 test    al, al
.text:00B155A5                 jz      loc_B15489
.text:00B155AB                 mov     [ebp+var_24], 1
.text:00B155B2                 jmp     loc_B15489
.text:00B155B7 ; ───────────────────────────────────────────────────────────────────────────
.text:00B155B7
.text:00B155B7 loc_B155B7:                             ; CODE XREF: sub_B15240+2D↑j
.text:00B155B7                                         ; sub_B15240+46↑j ...
```

```
.text:00B155B7                 pop     edi
.text:00B155B8                 pop     esi
.text:00B155B9                 xor     eax, eax
.text:00B155BB                 pop     ebx
.text:00B155BC                 mov     esp, ebp
.text:00B155BE                 pop     ebp
.text:00B155BF                 retn
.text:00B155BF sub_B15240      endp
.text:00B155BF
.text:00B155BF ; ───────────────────────────────────────────────────────────────────────────
.text:00B155C0                 db 0E0h dup(0CCh)
.text:00B156A0
.text:00B156A0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B156A0
.text:00B156A0 ; Attributes: bp-based frame
.text:00B156A0
.text:00B156A0 SEH_4132B0      proc near               ; DATA XREF: sub_B132B0+A↑o
.text:00B156A0
.text:00B156A0 var_4           = dword ptr -4
.text:00B156A0 arg_0           = dword ptr  8
.text:00B156A0 arg_4           = dword ptr  0Ch
.text:00B156A0 arg_8           = dword ptr  10h
.text:00B156A0 arg_C           = dword ptr  14h
.text:00B156A0
.text:00B156A0                 push    ebp
.text:00B156A1                 mov     ebp, esp
.text:00B156A3                 push    ecx
.text:00B156A4                 mov     eax, [ebp+arg_C]
.text:00B156A7                 push    eax
.text:00B156A8                 mov     ecx, [ebp+arg_8]
.text:00B156AB                 push    ecx
.text:00B156AC                 mov     edx, [ebp+arg_4]
.text:00B156AF                 push    edx
.text:00B156B0                 mov     eax, [ebp+arg_0]
.text:00B156B3                 push    eax
.text:00B156B4                 call    SEH_4128D0
.text:00B156B9                 add     esp, 10h
.text:00B156BC                 mov     [ebp+var_4], eax
.text:00B156BF                 mov     ecx, [ebp+arg_0]
.text:00B156C2                 mov     edx, [ecx+4]
.text:00B156C5                 and     edx, 66h
.text:00B156C8                 jnz     short loc_B156F4
.text:00B156CA                 mov     eax, [ebp+arg_0]
.text:00B156CD                 cmp     dword ptr [eax], 0E06D7363h
.text:00B156D3                 jnz     short loc_B156F4
.text:00B156D5                 cmp     [ebp+var_4], 1
.text:00B156D9                 jnz     short loc_B156F4
.text:00B156DB                 call    j___current_exception
.text:00B156E0                 mov     ecx, [ebp+arg_0]
.text:00B156E3                 mov     [eax], ecx
.text:00B156E5                 call    j___current_exception_context
.text:00B156EA                 mov     edx, [ebp+arg_8]
.text:00B156ED                 mov     [eax], edx
.text:00B156EF                 call    j_terminate
.text:00B156F4
.text:00B156F4 loc_B156F4:                             ; CODE XREF: SEH_4132B0+28↑j
.text:00B156F4                                         ; SEH_4132B0+33↑j  …
.text:00B156F4                 mov     eax, [ebp+var_4]
.text:00B156F7                 mov     esp, ebp
.text:00B156F9                 pop     ebp
.text:00B156FA                 retn
.text:00B156FA SEH_4132B0      endp
.text:00B156FA
.text:00B156FA ; ───────────────────────────────────────────────────────────────────────────
.text:00B156FB                 db 25h dup(0CCh)
.text:00B15720
.text:00B15720 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B15720
.text:00B15720 ; Attributes: bp-based frame
.text:00B15720
.text:00B15720 sub_B15720      proc near               ; CODE XREF: sub_B111EA↑j
.text:00B15720
.text:00B15720 var_48          = dword ptr -48h
.text:00B15720 var_44          = dword ptr -44h
.text:00B15720 var_40          = dword ptr -40h
```

```
.text:00B15720 var_3C          = dword ptr -3Ch
.text:00B15720 var_38          = dword ptr -38h
.text:00B15720 var_34          = dword ptr -34h
.text:00B15720 var_30          = dword ptr -30h
.text:00B15720 var_2C          = dword ptr -2Ch
.text:00B15720 var_28          = dword ptr -28h
.text:00B15720 var_21          = byte ptr -21h
.text:00B15720 var_20          = dword ptr -20h
.text:00B15720 var_10          = dword ptr -10h
.text:00B15720 var_C           = dword ptr -0Ch
.text:00B15720 var_8           = dword ptr -8
.text:00B15720 var_4           = dword ptr -4
.text:00B15720
.text:00B15720                 push    ebp
.text:00B15721                 mov     ebp, esp
.text:00B15723                 sub     esp, 48h
.text:00B15726                 mov     eax, ___security_cookie
.text:00B1572B                 xor     eax, ebp
.text:00B1572D                 mov     [ebp+var_4], eax
.text:00B15730                 push    ebx
.text:00B15731                 push    esi
.text:00B15732                 mov     dword_B1C574, 0
.text:00B1573C                 mov     eax, dword_B1C02C
.text:00B15741                 or      eax, 1
.text:00B15744                 mov     dword_B1C02C, eax
.text:00B15749                 push    0Ah             ; ProcessorFeature
.text:00B1574B                 call    ds:IsProcessorFeaturePresent
.text:00B15751                 test    eax, eax
.text:00B15753                 jnz     short loc_B1575C
.text:00B15755                 xor     eax, eax
.text:00B15757                 jmp     loc_B15A43
.text:00B1575C ; ---------------------------------------------------------------------------
.text:00B1575C
.text:00B1575C loc_B1575C:                             ; CODE XREF: sub_B15720+33↑j
.text:00B1575C                 mov     [ebp+var_10], 0
.text:00B15763                 mov     [ebp+var_C], 0
.text:00B1576A                 mov     [ebp+var_8], 0
.text:00B15771                 lea     esi, [ebp+var_20]
.text:00B15774                 xor     eax, eax
.text:00B15776                 xor     ecx, ecx
.text:00B15778                 cpuid
.text:00B1577A                 mov     [esi], eax
.text:00B1577C                 mov     [esi+4], ebx
.text:00B1577F                 mov     [esi+8], ecx
.text:00B15782                 mov     [esi+0Ch], edx
.text:00B15785                 mov     ecx, 4
.text:00B1578A                 imul    edx, ecx, 0
.text:00B1578D                 mov     eax, [ebp+edx+var_20]
.text:00B15791                 mov     [ebp+var_38], eax
.text:00B15794                 mov     ecx, 4
.text:00B15799                 shl     ecx, 0
.text:00B1579C                 mov     edx, [ebp+ecx+var_20]
.text:00B157A0                 xor     edx, 756E6547h
.text:00B157A6                 mov     eax, 4
.text:00B157AB                 imul    ecx, eax, 3
.text:00B157AE                 mov     eax, [ebp+ecx+var_20]
.text:00B157B2                 xor     eax, 49656E69h
.text:00B157B7                 or      edx, eax
.text:00B157B9                 mov     ecx, 4
.text:00B157BE                 shl     ecx, 1
.text:00B157C0                 mov     eax, [ebp+ecx+var_20]
.text:00B157C4                 xor     eax, 6C65746Eh
.text:00B157C9                 or      edx, eax
.text:00B157CB                 jnz     short loc_B157D6
.text:00B157CD                 mov     [ebp+var_2C], 1
.text:00B157D4                 jmp     short loc_B157DD
.text:00B157D6 ; ---------------------------------------------------------------------------
.text:00B157D6
.text:00B157D6 loc_B157D6:                             ; CODE XREF: sub_B15720+AB↑j
.text:00B157D6                 mov     [ebp+var_2C], 0
.text:00B157DD
.text:00B157DD loc_B157DD:                             ; CODE XREF: sub_B15720+B4↑j
.text:00B157DD                 mov     cl, byte ptr [ebp+var_2C]
.text:00B157E0                 mov     [ebp+var_21], cl
.text:00B157E3                 lea     esi, [ebp+var_20]
```

```
.text:00B157E6                 mov     eax, 1
.text:00B157EB                 xor     ecx, ecx
.text:00B157ED                 cpuid
.text:00B157EF                 mov     [esi], eax
.text:00B157F1                 mov     [esi+4], ebx
.text:00B157F4                 mov     [esi+8], ecx
.text:00B157F7                 mov     [esi+0Ch], edx
.text:00B157FA                 mov     edx, 4
.text:00B157FF                 imul    eax, edx, 0
.text:00B15802                 mov     ecx, [ebp+eax+var_20]
.text:00B15806                 mov     [ebp+var_28], ecx
.text:00B15809                 movzx   edx, [ebp+var_21]
.text:00B1580D                 test    edx, edx
.text:00B1580F                 jz      short loc_B15880
.text:00B15811                 mov     eax, [ebp+var_28]
.text:00B15814                 and     eax, 0FFF3FF0h
.text:00B15819                 cmp     eax, 106C0h
.text:00B1581E                 jz      short loc_B15873
.text:00B15820                 mov     ecx, [ebp+var_28]
.text:00B15823                 and     ecx, 0FFF3FF0h
.text:00B15829                 cmp     ecx, 20660h
.text:00B1582F                 jz      short loc_B15873
.text:00B15831                 mov     edx, [ebp+var_28]
.text:00B15834                 and     edx, 0FFF3FF0h
.text:00B1583A                 cmp     edx, 20670h
.text:00B15840                 jz      short loc_B15873
.text:00B15842                 mov     eax, [ebp+var_28]
.text:00B15845                 and     eax, 0FFF3FF0h
.text:00B1584A                 cmp     eax, 30650h
.text:00B1584F                 jz      short loc_B15873
.text:00B15851                 mov     ecx, [ebp+var_28]
.text:00B15854                 and     ecx, 0FFF3FF0h
.text:00B1585A                 cmp     ecx, 30660h
.text:00B15860                 jz      short loc_B15873
.text:00B15862                 mov     edx, [ebp+var_28]
.text:00B15865                 and     edx, 0FFF3FF0h
.text:00B1586B                 cmp     edx, 30670h
.text:00B15871                 jnz     short loc_B15880
.text:00B15873
.text:00B15873 loc_B15873:                             ; CODE XREF: sub_B15720+FE↑j
.text:00B15873                                         ; sub_B15720+10F↑j ...
.text:00B15873                 mov     eax, dword_B1C578
.text:00B15878                 or      eax, 1
.text:00B1587B                 mov     dword_B1C578, eax
.text:00B15880
.text:00B15880 loc_B15880:                             ; CODE XREF: sub_B15720+EF↑j
.text:00B15880                                         ; sub_B15720+151↑j
.text:00B15880                 mov     ecx, 4
.text:00B15885                 imul    edx, ecx, 3
.text:00B15888                 mov     eax, 4
.text:00B1588D                 imul    ecx, eax, 0
.text:00B15890                 mov     edx, [ebp+edx+var_20]
.text:00B15894                 mov     [ebp+ecx+var_10], edx
.text:00B15898                 mov     eax, 4
.text:00B1589D                 shl     eax, 1
.text:00B1589F                 mov     ecx, 4
.text:00B158A4                 shl     ecx, 0
.text:00B158A7                 mov     edx, [ebp+eax+var_20]
.text:00B158AB                 mov     [ebp+ecx+var_10], edx
.text:00B158AF                 cmp     [ebp+var_38], 7
.text:00B158B3                 jl      short loc_B15906
.text:00B158B5                 lea     esi, [ebp+var_20]
.text:00B158B8                 mov     eax, 7
.text:00B158BD                 xor     ecx, ecx
.text:00B158BF                 cpuid
.text:00B158C1                 mov     [esi], eax
.text:00B158C3                 mov     [esi+4], ebx
.text:00B158C6                 mov     [esi+8], ecx
.text:00B158C9                 mov     [esi+0Ch], edx
.text:00B158CC                 mov     eax, 4
.text:00B158D1                 shl     eax, 0
.text:00B158D4                 mov     ecx, 4
.text:00B158D9                 shl     ecx, 1
.text:00B158DB                 mov     edx, [ebp+eax+var_20]
.text:00B158DF                 mov     [ebp+ecx+var_10], edx
```

```
.text:00B158E3                 mov     eax, 4
.text:00B158E8                 shl     eax, 0
.text:00B158EB                 mov     ecx, [ebp+eax+var_20]
.text:00B158EF                 and     ecx, 200h
.text:00B158F5                 jz      short loc_B15906
.text:00B158F7                 mov     edx, dword_B1C578
.text:00B158FD                 or      edx, 2
.text:00B15900                 mov     dword_B1C578, edx
.text:00B15906
.text:00B15906 loc_B15906:                             ; CODE XREF: sub_B15720+193↑j
.text:00B15906                                         ; sub_B15720+1D5↑j
.text:00B15906                 mov     dword_B1C574, 1
.text:00B15910                 mov     eax, dword_B1C02C
.text:00B15915                 or      eax, 2
.text:00B15918                 mov     dword_B1C02C, eax
.text:00B1591D                 mov     ecx, 4
.text:00B15922                 shl     ecx, 0
.text:00B15925                 mov     edx, [ebp+ecx+var_10]
.text:00B15929                 and     edx, 100000h
.text:00B1592F                 jz      loc_B15A41
.text:00B15935                 mov     dword_B1C574, 2
.text:00B1593F                 mov     eax, dword_B1C02C
.text:00B15944                 or      eax, 4
.text:00B15947                 mov     dword_B1C02C, eax
.text:00B1594C                 mov     ecx, 4
.text:00B15951                 shl     ecx, 0
.text:00B15954                 mov     edx, [ebp+ecx+var_10]
.text:00B15958                 and     edx, 8000000h
.text:00B1595E                 jz      loc_B15A41
.text:00B15964                 mov     eax, 4
.text:00B15969                 shl     eax, 0
.text:00B1596C                 mov     ecx, [ebp+eax+var_10]
.text:00B15970                 and     ecx, 10000000h
.text:00B15976                 jz      loc_B15A41
.text:00B1597C                 xor     ecx, ecx
.text:00B1597E                 xgetbv
.text:00B15981                 mov     [ebp+var_34], eax
.text:00B15984                 mov     [ebp+var_30], edx
.text:00B15987                 mov     edx, [ebp+var_34]
.text:00B1598A                 mov     eax, [ebp+var_30]
.text:00B1598D                 and     edx, 6
.text:00B15990                 and     eax, 0
.text:00B15993                 mov     [ebp+var_40], edx
.text:00B15996                 mov     [ebp+var_3C], eax
.text:00B15999                 cmp     [ebp+var_40], 6
.text:00B1599D                 jnz     loc_B15A41
.text:00B159A3                 cmp     [ebp+var_3C], 0
.text:00B159A7                 jnz     loc_B15A41
.text:00B159AD                 mov     dword_B1C574, 3
.text:00B159B7                 mov     ecx, dword_B1C02C
.text:00B159BD                 or      ecx, 8
.text:00B159C0                 mov     dword_B1C02C, ecx
.text:00B159C6                 mov     edx, 4
.text:00B159CB                 shl     edx, 1
.text:00B159CD                 mov     eax, [ebp+edx+var_10]
.text:00B159D1                 and     eax, 20h
.text:00B159D4                 jz      short loc_B15A41
.text:00B159D6                 mov     dword_B1C574, 5
.text:00B159E0                 mov     ecx, dword_B1C02C
.text:00B159E6                 or      ecx, 20h
.text:00B159E9                 mov     dword_B1C02C, ecx
.text:00B159EF                 mov     edx, 4
.text:00B159F4                 shl     edx, 1
.text:00B159F6                 mov     eax, [ebp+edx+var_10]
.text:00B159FA                 and     eax, 0D0030000h
.text:00B159FF                 cmp     eax, 0D0030000h
.text:00B15A04                 jnz     short loc_B15A41
.text:00B15A06                 mov     ecx, [ebp+var_34]
.text:00B15A09                 mov     edx, [ebp+var_30]
.text:00B15A0C                 and     ecx, 0E0h
.text:00B15A12                 and     edx, 0
.text:00B15A15                 mov     [ebp+var_48], ecx
.text:00B15A18                 mov     [ebp+var_44], edx
.text:00B15A1B                 cmp     [ebp+var_48], 0E0h
.text:00B15A22                 jnz     short loc_B15A41
```

```
.text:00B15A24                 cmp     [ebp+var_44], 0
.text:00B15A28                 jnz     short loc_B15A41
.text:00B15A2A                 mov     dword_B1C574, 6
.text:00B15A34                 mov     eax, dword_B1C02C
.text:00B15A39                 or      eax, 40h
.text:00B15A3C                 mov     dword_B1C02C, eax
.text:00B15A41
.text:00B15A41 loc_B15A41:                             ; CODE XREF: sub_B15720+20F↑j
.text:00B15A41                                         ; sub_B15720+23E↑j ...
.text:00B15A41                 xor     eax, eax
.text:00B15A43
.text:00B15A43 loc_B15A43:                             ; CODE XREF: sub_B15720+37↑j
.text:00B15A43                 pop     esi
.text:00B15A44                 pop     ebx
.text:00B15A45                 mov     ecx, [ebp+var_4]
.text:00B15A48                 xor     ecx, ebp        ; StackCookie
.text:00B15A4A                 call    j_@__security_check_cookie@4 ; __security_check_cookie(x)
.text:00B15A4F                 mov     esp, ebp
.text:00B15A51                 pop     ebp
.text:00B15A52                 retn
.text:00B15A52 sub_B15720      endp
.text:00B15A52
.text:00B15A52 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15A53                 db 0CDh dup(0CCh)
.text:00B15B20
.text:00B15B20 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B15B20
.text:00B15B20 ; Attributes: bp-based frame
.text:00B15B20
.text:00B15B20 sub_B15B20      proc near               ; CODE XREF: sub_B111E5↑j
.text:00B15B20
.text:00B15B20 var_4           = dword ptr -4
.text:00B15B20
.text:00B15B20                 push    ebp
.text:00B15B21                 mov     ebp, esp
.text:00B15B23                 push    ecx
.text:00B15B24                 cmp     dword_B1C030, 0
.text:00B15B2B                 jz      short loc_B15B36
.text:00B15B2D                 mov     [ebp+var_4], 1
.text:00B15B34                 jmp     short loc_B15B3D
.text:00B15B36 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15B36
.text:00B15B36 loc_B15B36:                             ; CODE XREF: sub_B15B20+B↑j
.text:00B15B36                 mov     [ebp+var_4], 0
.text:00B15B3D
.text:00B15B3D loc_B15B3D:                             ; CODE XREF: sub_B15B20+14↑j
.text:00B15B3D                 mov     eax, [ebp+var_4]
.text:00B15B40                 mov     esp, ebp
.text:00B15B42                 pop     ebp
.text:00B15B43                 retn
.text:00B15B43 sub_B15B20      endp
.text:00B15B43
.text:00B15B43 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15B44                 db 9 dup(0CCh)
.text:00B15B4D ; [00000006 BYTES: COLLAPSED FUNCTION __CxxFrameHandler3. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B53 ; [00000006 BYTES: COLLAPSED FUNCTION __std_type_info_destroy_list. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B59 ; [00000006 BYTES: COLLAPSED FUNCTION __current_exception. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B5F ; [00000006 BYTES: COLLAPSED FUNCTION __current_exception_context. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B65 ; [00000006 BYTES: COLLAPSED FUNCTION memset. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B6B ; [00000006 BYTES: COLLAPSED FUNCTION _except_handler4_common. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B71 ; [00000006 BYTES: COLLAPSED FUNCTION __vcrt_GetModuleFileNameW. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B77 ; [00000006 BYTES: COLLAPSED FUNCTION __vcrt_GetModuleHandleW. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B7D ; [00000006 BYTES: COLLAPSED FUNCTION __vcrt_LoadLibraryExW. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B83 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15B83
.text:00B15B83 loc_B15B83:                             ; CODE XREF: .text:00B113D9↑j
.text:00B15B83                 jmp     ds:strlen
.text:00B15B89 ; [00000006 BYTES: COLLAPSED FUNCTION _CrtDbgReport. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B8F ; [00000006 BYTES: COLLAPSED FUNCTION _CrtDbgReportW. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B95 ; [00000006 BYTES: COLLAPSED FUNCTION _seh_filter_exe. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15B9B ; [00000006 BYTES: COLLAPSED FUNCTION _set_app_type. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BA1 ; [00000006 BYTES: COLLAPSED FUNCTION __setusermatherr. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BA7 ; [00000006 BYTES: COLLAPSED FUNCTION _configure_narrow_argv. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BAD ; [00000006 BYTES: COLLAPSED FUNCTION _initialize_narrow_environment. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BB3 ; [00000006 BYTES: COLLAPSED FUNCTION _get_initial_narrow_environment. PRESS CTRL-NUMPAD+ TO EXPAND]
```

```
.text:00B15BB9 ; [00000006 BYTES: COLLAPSED FUNCTION _initterm. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BBF ; [00000006 BYTES: COLLAPSED FUNCTION _initterm_e. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BC5 ; [00000006 BYTES: COLLAPSED FUNCTION exit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BCB ; [00000006 BYTES: COLLAPSED FUNCTION _exit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BD1 ; [00000006 BYTES: COLLAPSED FUNCTION _set_fmode. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BD7 ; [00000006 BYTES: COLLAPSED FUNCTION __p___argc. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BDD ; [00000006 BYTES: COLLAPSED FUNCTION __p___argv. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BE3 ; [00000006 BYTES: COLLAPSED FUNCTION _cexit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BE9 ; [00000006 BYTES: COLLAPSED FUNCTION _c_exit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BEF ; [00000006 BYTES: COLLAPSED FUNCTION _register_thread_local_exe_atexit_callback. PRESS CTRL-NUMPAD+
TO EXPAND]
.text:00B15BF5 ; [00000006 BYTES: COLLAPSED FUNCTION _configthreadlocale. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15BFB ; [00000006 BYTES: COLLAPSED FUNCTION _set_new_mode. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C01 ; [00000006 BYTES: COLLAPSED FUNCTION __p__commode. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C07 ; [00000006 BYTES: COLLAPSED FUNCTION strcpy_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C0D ; [00000006 BYTES: COLLAPSED FUNCTION strcat_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C13 ; [00000006 BYTES: COLLAPSED FUNCTION __stdio_common_vsprintf_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C19 ; [00000006 BYTES: COLLAPSED FUNCTION _seh_filter_dll. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C1F ; [00000006 BYTES: COLLAPSED FUNCTION _initialize_onexit_table. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C25 ; [00000006 BYTES: COLLAPSED FUNCTION _register_onexit_function. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C2B ; [00000006 BYTES: COLLAPSED FUNCTION _execute_onexit_table. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C31 ; [00000006 BYTES: COLLAPSED FUNCTION _crt_atexit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C37 ; [00000006 BYTES: COLLAPSED FUNCTION _crt_at_quick_exit. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C3D ; [00000006 BYTES: COLLAPSED FUNCTION _controlfp_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C43 ; [00000006 BYTES: COLLAPSED FUNCTION terminate. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C49 ; [00000006 BYTES: COLLAPSED FUNCTION _wmakepath_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C4F ; [00000006 BYTES: COLLAPSED FUNCTION _wsplitpath_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C55 ; [00000006 BYTES: COLLAPSED FUNCTION wcscpy_s. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00B15C5B ; ---------------------------------------------------------------------------
.text:00B15C5B
.text:00B15C5B loc_B15C5B:                             ; CODE XREF: .text:00B110F5↑j
.text:00B15C5B                 jmp     ds:GetCurrentThreadId
.text:00B15C61 ; ---------------------------------------------------------------------------
.text:00B15C61
.text:00B15C61 loc_B15C61:                             ; CODE XREF: .text:00B111B3↑j
.text:00B15C61                 jmp     ds:IsDebuggerPresent
.text:00B15C67 ; ---------------------------------------------------------------------------
.text:00B15C67
.text:00B15C67 loc_B15C67:                             ; CODE XREF: .text:00B11361↑j
.text:00B15C67                 jmp     ds:RaiseException
.text:00B15C6D ; ---------------------------------------------------------------------------
.text:00B15C6D
.text:00B15C6D loc_B15C6D:                             ; CODE XREF: .text:00B111CC↑j
.text:00B15C6D                 jmp     ds:MultiByteToWideChar
.text:00B15C73 ; ---------------------------------------------------------------------------
.text:00B15C73
.text:00B15C73 loc_B15C73:                             ; CODE XREF: .text:00B11285↑j
.text:00B15C73                 jmp     ds:WideCharToMultiByte
.text:00B15C79 ; ---------------------------------------------------------------------------
.text:00B15C79
.text:00B15C79 loc_B15C79:                             ; CODE XREF: .text:00B11226↑j
.text:00B15C79                 jmp     ds:UnhandledExceptionFilter
.text:00B15C7F ; ---------------------------------------------------------------------------
.text:00B15C7F
.text:00B15C7F loc_B15C7F:                             ; CODE XREF: .text:00B11028↑j
.text:00B15C7F                 jmp     ds:SetUnhandledExceptionFilter
.text:00B15C85 ; ---------------------------------------------------------------------------
.text:00B15C85
.text:00B15C85 loc_B15C85:                             ; CODE XREF: .text:00B1137F↑j
.text:00B15C85                 jmp     ds:GetCurrentProcess
.text:00B15C8B ; ---------------------------------------------------------------------------
.text:00B15C8B
.text:00B15C8B loc_B15C8B:                             ; CODE XREF: .text:00B1123F↑j
.text:00B15C8B                 jmp     ds:TerminateProcess
.text:00B15C91 ; ---------------------------------------------------------------------------
.text:00B15C91
.text:00B15C91 loc_B15C91:                             ; CODE XREF: .text:00B11212↑j
.text:00B15C91                 jmp     ds:IsProcessorFeaturePresent
.text:00B15C97 ; ---------------------------------------------------------------------------
.text:00B15C97
.text:00B15C97 loc_B15C97:                             ; CODE XREF: .text:00B111BD↑j
.text:00B15C97                 jmp     ds:QueryPerformanceCounter
.text:00B15C9D ; ---------------------------------------------------------------------------
.text:00B15C9D
.text:00B15C9D loc_B15C9D:                             ; CODE XREF: .text:00B11109↑j
```

```
.text:00B15C9D                 jmp     ds:GetCurrentProcessId
.text:00B15CA3 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CA3
.text:00B15CA3 loc_B15CA3:                             ; CODE XREF: .text:00B11041↑j
.text:00B15CA3                 jmp     ds:GetSystemTimeAsFileTime
.text:00B15CA9 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CA9
.text:00B15CA9 loc_B15CA9:                             ; CODE XREF: .text:00B11406↑j
.text:00B15CA9                 jmp     ds:InitializeSListHead
.text:00B15CAF ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CAF
.text:00B15CAF loc_B15CAF:                             ; CODE XREF: .text:00B1139D↑j
.text:00B15CAF                 jmp     ds:GetStartupInfoW
.text:00B15CB5 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CB5
.text:00B15CB5 loc_B15CB5:                             ; CODE XREF: .text:00B11140↑j
.text:00B15CB5                 jmp     ds:GetModuleHandleW
.text:00B15CBB ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CBB
.text:00B15CBB loc_B15CBB:                             ; CODE XREF: .text:00B1131B↑j
.text:00B15CBB                 jmp     ds:GetLastError
.text:00B15CC1 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CC1
.text:00B15CC1 loc_B15CC1:                             ; CODE XREF: .text:00B11014↑j
.text:00B15CC1                 jmp     ds:HeapAlloc
.text:00B15CC7 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CC7
.text:00B15CC7 loc_B15CC7:                             ; CODE XREF: .text:00B1142E↑j
.text:00B15CC7                 jmp     ds:HeapFree
.text:00B15CCD ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CCD
.text:00B15CCD loc_B15CCD:                             ; CODE XREF: .text:00B110F0↑j
.text:00B15CCD                 jmp     ds:GetProcessHeap
.text:00B15CD3 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CD3
.text:00B15CD3 loc_B15CD3:                             ; CODE XREF: .text:00B111F9↑j
.text:00B15CD3                 jmp     ds:VirtualQuery
.text:00B15CD9 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CD9
.text:00B15CD9 loc_B15CD9:                             ; CODE XREF: .text:00B1110E↑j
.text:00B15CD9                 jmp     ds:FreeLibrary
.text:00B15CDF ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CDF
.text:00B15CDF loc_B15CDF:                             ; CODE XREF: .text:00B111E0↑j
.text:00B15CDF                 jmp     ds:GetProcAddress
.text:00B15CDF ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CE5                 align 10h
.text:00B15CF0
.text:00B15CF0 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B15CF0
.text:00B15CF0 ; Attributes: bp-based frame
.text:00B15CF0
.text:00B15CF0 sub_B15CF0      proc near               ; CODE XREF: .text:00B112B7↑j
.text:00B15CF0                                         ; sub_B112DA↑j ...
.text:00B15CF0                 push    ebp
.text:00B15CF1                 mov     ebp, esp
.text:00B15CF3                 mov     al, 1
.text:00B15CF5                 pop     ebp
.text:00B15CF6                 retn
.text:00B15CF6 sub_B15CF0      endp
.text:00B15CF6
.text:00B15CF6 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15CF7                 align 10h
.text:00B15D00
.text:00B15D00 ; ═══════════════ S U B R O U T I N E ═══════════════════════════════════════
.text:00B15D00
.text:00B15D00 ; Attributes: bp-based frame
.text:00B15D00
.text:00B15D00 sub_B15D00      proc near               ; CODE XREF: sub_B11299↑j
.text:00B15D00                                         ; sub_B112C6↑j ...
.text:00B15D00                 push    ebp
.text:00B15D01                 mov     ebp, esp
.text:00B15D03                 mov     al, 1
.text:00B15D05                 pop     ebp
.text:00B15D06                 retn
```

```
.text:00B15D06 sub_B15D00      endp
.text:00B15D06
.text:00B15D06 ; ---------------------------------------------------------------------------
.text:00B15D07                 align 10h
.text:00B15D10
.text:00B15D10 ; =============== S U B R O U T I N E =======================================
.text:00B15D10
.text:00B15D10 ; Attributes: bp-based frame
.text:00B15D10
.text:00B15D10 sub_B15D10      proc near               ; CODE XREF: sub_B11136↑j
.text:00B15D10                                         ; sub_B111D1↑j ...
.text:00B15D10                 push    ebp
.text:00B15D11                 mov     ebp, esp
.text:00B15D13                 mov     al, 1
.text:00B15D15                 pop     ebp
.text:00B15D16                 retn
.text:00B15D16 sub_B15D10      endp
.text:00B15D16
.text:00B15D16 ; ---------------------------------------------------------------------------
.text:00B15D17                 align 10h
.text:00B15D20
.text:00B15D20 ; =============== S U B R O U T I N E =======================================
.text:00B15D20
.text:00B15D20 ; Attributes: bp-based frame
.text:00B15D20
.text:00B15D20 sub_B15D20      proc near               ; CODE XREF: sub_B1101E↑j
.text:00B15D20                                         ; .text:00B110C3↑j ...
.text:00B15D20                 push    ebp
.text:00B15D21                 mov     ebp, esp
.text:00B15D23                 mov     al, 1
.text:00B15D25                 pop     ebp
.text:00B15D26                 retn
.text:00B15D26 sub_B15D20      endp
.text:00B15D26
.text:00B15D26 ; ---------------------------------------------------------------------------
.text:00B15D27                 align 10h
.text:00B15D30
.text:00B15D30 ; =============== S U B R O U T I N E =======================================
.text:00B15D30
.text:00B15D30 ; Attributes: bp-based frame
.text:00B15D30
.text:00B15D30 sub_B15D30      proc near               ; CODE XREF: .text:00B111EF↑j
.text:00B15D30                                         ; sub_B112A8↑j ...
.text:00B15D30                 push    ebp
.text:00B15D31                 mov     ebp, esp
.text:00B15D33                 mov     al, 1
.text:00B15D35                 pop     ebp
.text:00B15D36                 retn
.text:00B15D36 sub_B15D30      endp
.text:00B15D36
.text:00B15D36 ; ---------------------------------------------------------------------------
.text:00B15D37                 align 10h
.text:00B15D40
.text:00B15D40 ; =============== S U B R O U T I N E =======================================
.text:00B15D40
.text:00B15D40 ; Attributes: bp-based frame
.text:00B15D40
.text:00B15D40 sub_B15D40      proc near               ; CODE XREF: .text:00B11082↑j
.text:00B15D40                                         ; sub_B11235↑j
.text:00B15D40                 push    ebp
.text:00B15D41                 mov     ebp, esp
.text:00B15D43                 xor     eax, eax
.text:00B15D45                 pop     ebp
.text:00B15D46                 retn
.text:00B15D46 sub_B15D40      endp
.text:00B15D46
.text:00B15D46 ; ---------------------------------------------------------------------------
.text:00B15D47                 align 10h
.text:00B15D50
.text:00B15D50 ; =============== S U B R O U T I N E =======================================
.text:00B15D50
.text:00B15D50 ; Attributes: bp-based frame
.text:00B15D50
.text:00B15D50 sub_B15D50      proc near               ; CODE XREF: .text:00B1128A↑j
.text:00B15D50                                         ; sub_B1140B↑j
```

```
.text:00B15D50
.text:00B15D50 var_54          = dword ptr -54h
.text:00B15D50 var_50          = dword ptr -50h
.text:00B15D50 var_4C          = dword ptr -4Ch
.text:00B15D50 var_48          = dword ptr -48h
.text:00B15D50 var_44          = dword ptr -44h
.text:00B15D50 var_40          = dword ptr -40h
.text:00B15D50 var_3C          = dword ptr -3Ch
.text:00B15D50 var_38          = dword ptr -38h
.text:00B15D50 var_34          = dword ptr -34h
.text:00B15D50 var_30          = dword ptr -30h
.text:00B15D50 var_2C          = dword ptr -2Ch
.text:00B15D50 var_28          = dword ptr -28h
.text:00B15D50 var_24          = dword ptr -24h
.text:00B15D50 var_20          = dword ptr -20h
.text:00B15D50 var_1C          = dword ptr -1Ch
.text:00B15D50 var_18          = dword ptr -18h
.text:00B15D50 var_14          = dword ptr -14h
.text:00B15D50 var_10          = dword ptr -10h
.text:00B15D50 var_C           = dword ptr -0Ch
.text:00B15D50 var_8           = dword ptr -8
.text:00B15D50 var_4           = dword ptr -4
.text:00B15D50 arg_0           = dword ptr  8
.text:00B15D50
.text:00B15D50                 push    ebp
.text:00B15D51                 mov     ebp, esp
.text:00B15D53                 sub     esp, 54h
.text:00B15D56                 cmp     dword_B1C574, 1
.text:00B15D5D                 jge     short loc_B15D67
.text:00B15D5F                 mov     eax, [ebp+arg_0]
.text:00B15D62                 jmp     loc_B15E80
.text:00B15D67 ; ---------------------------------------------------------------------------
.text:00B15D67
.text:00B15D67 loc_B15D67:                             ; CODE XREF: sub_B15D50+D↑j
.text:00B15D67                 cmp     [ebp+arg_0], 0C00002B4h
.text:00B15D6E                 jz      short loc_B15D81
.text:00B15D70                 cmp     [ebp+arg_0], 0C00002B5h
.text:00B15D77                 jz      short loc_B15D81
.text:00B15D79                 mov     eax, [ebp+arg_0]
.text:00B15D7C                 jmp     loc_B15E80
.text:00B15D81 ; ---------------------------------------------------------------------------
.text:00B15D81
.text:00B15D81 loc_B15D81:                             ; CODE XREF: sub_B15D50+1E↑j
.text:00B15D81                                         ; sub_B15D50+27↑j
.text:00B15D81                 mov     [ebp+var_C], 1
.text:00B15D88                 mov     [ebp+var_10], 2
.text:00B15D8F                 mov     [ebp+var_14], 4
.text:00B15D96                 mov     [ebp+var_18], 8
.text:00B15D9D                 mov     [ebp+var_1C], 10h
.text:00B15DA4                 mov     [ebp+var_20], 20h ; ' '
.text:00B15DAB                 mov     [ebp+var_24], 3Fh ; '?'
.text:00B15DB2                 mov     [ebp+var_28], 80h
.text:00B15DB9                 mov     [ebp+var_2C], 100h
.text:00B15DC0                 mov     [ebp+var_30], 200h
.text:00B15DC7                 mov     [ebp+var_34], 400h
.text:00B15DCE                 mov     [ebp+var_38], 800h
.text:00B15DD5                 mov     [ebp+var_3C], 1000h
.text:00B15DDC                 mov     [ebp+var_40], 81h
.text:00B15DE3                 mov     [ebp+var_44], 102h
.text:00B15DEA                 mov     [ebp+var_48], 204h
.text:00B15DF1                 mov     [ebp+var_4C], 408h
.text:00B15DF8                 mov     [ebp+var_50], 810h
.text:00B15DFF                 mov     [ebp+var_54], 1020h
.text:00B15E06                 stmxcsr [ebp+var_8]
.text:00B15E0A                 mov     eax, [ebp+var_8]
.text:00B15E0D                 xor     eax, 3Fh
.text:00B15E10                 mov     [ebp+var_4], eax
.text:00B15E13                 mov     ecx, [ebp+var_4]
.text:00B15E16                 and     ecx, 81h
.text:00B15E1C                 jnz     short loc_B15E25
.text:00B15E1E                 mov     eax, 0C0000090h
.text:00B15E23                 jmp     short loc_B15E80
.text:00B15E25 ; ---------------------------------------------------------------------------
.text:00B15E25
.text:00B15E25 loc_B15E25:                             ; CODE XREF: sub_B15D50+CC↑j
```

```
.text:00B15E25                 mov     edx, [ebp+var_4]
.text:00B15E28                 and     edx, 204h
.text:00B15E2E                 jnz     short loc_B15E37
.text:00B15E30                 mov     eax, 0C000008Eh
.text:00B15E35                 jmp     short loc_B15E80
.text:00B15E37 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15E37
.text:00B15E37 loc_B15E37:                             ; CODE XREF: sub_B15D50+DE↑j
.text:00B15E37                 mov     eax, [ebp+var_4]
.text:00B15E3A                 and     eax, 102h
.text:00B15E3F                 jnz     short loc_B15E48
.text:00B15E41                 mov     eax, 0C0000090h
.text:00B15E46                 jmp     short loc_B15E80
.text:00B15E48 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15E48
.text:00B15E48 loc_B15E48:                             ; CODE XREF: sub_B15D50+EF↑j
.text:00B15E48                 mov     ecx, [ebp+var_4]
.text:00B15E4B                 and     ecx, 408h
.text:00B15E51                 jnz     short loc_B15E5A
.text:00B15E53                 mov     eax, 0C0000091h
.text:00B15E58                 jmp     short loc_B15E80
.text:00B15E5A ; ───────────────────────────────────────────────────────────────────────────
.text:00B15E5A
.text:00B15E5A loc_B15E5A:                             ; CODE XREF: sub_B15D50+101↑j
.text:00B15E5A                 mov     edx, [ebp+var_4]
.text:00B15E5D                 and     edx, 810h
.text:00B15E63                 jnz     short loc_B15E6C
.text:00B15E65                 mov     eax, 0C0000093h
.text:00B15E6A                 jmp     short loc_B15E80
.text:00B15E6C ; ───────────────────────────────────────────────────────────────────────────
.text:00B15E6C
.text:00B15E6C loc_B15E6C:                             ; CODE XREF: sub_B15D50+113↑j
.text:00B15E6C                 mov     eax, [ebp+var_4]
.text:00B15E6F                 and     eax, 1020h
.text:00B15E74                 jnz     short loc_B15E7D
.text:00B15E76                 mov     eax, 0C000008Fh
.text:00B15E7B                 jmp     short loc_B15E80
.text:00B15E7D ; ───────────────────────────────────────────────────────────────────────────
.text:00B15E7D
.text:00B15E7D loc_B15E7D:                             ; CODE XREF: sub_B15D50+124↑j
.text:00B15E7D                 mov     eax, [ebp+arg_0]
.text:00B15E80
.text:00B15E80 loc_B15E80:                             ; CODE XREF: sub_B15D50+12↑j
.text:00B15E80                                         ; sub_B15D50+2C↑j ...
.text:00B15E80                 mov     esp, ebp
.text:00B15E82                 pop     ebp
.text:00B15E83                 retn
.text:00B15E83 sub_B15D50      endp
.text:00B15E83
.text:00B15E83 ; ───────────────────────────────────────────────────────────────────────────
.text:00B15E84                 db 251Ah dup(0CCh)
.text:00B1839E                 align 80h
.text:00B18400                 dd 300h dup(?)
.text:00B18400 _text           ends
.text:00B18400
```