National Forensic Sciences University

School of Cyber security and Digital Forensics

Subject: CTMSCS S2 P3 Malware Analysis

(TA-II Assignment)

Submitted to: Mr. Dharmesh Dave and Mr. Parag Rughani

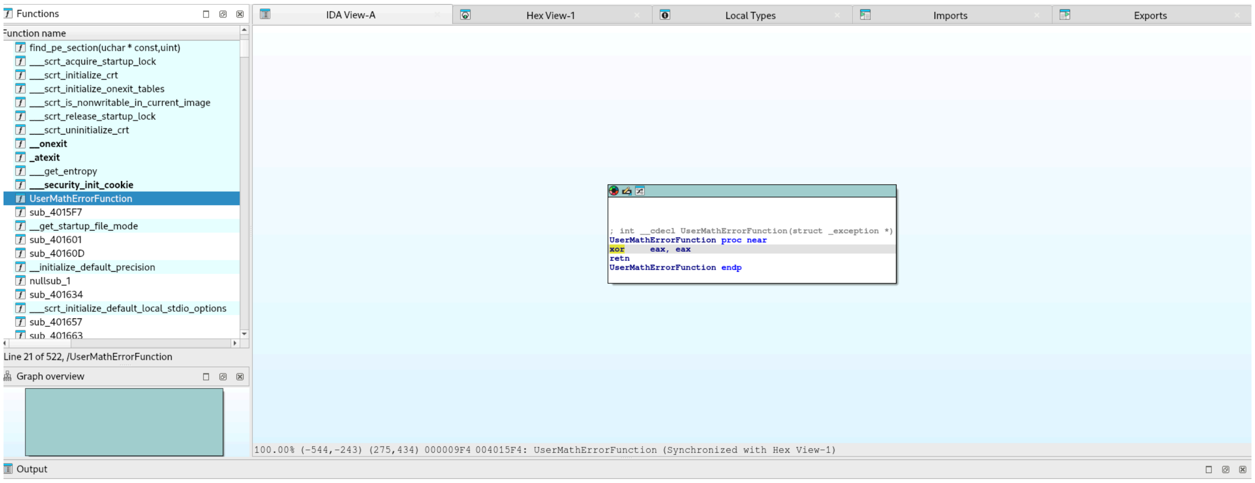
Submitted by: Disha Sharma (2401030020014)

Submission Date: 03/04/2025

INDEX

1. How many UDFs are there?
2. How many Libraries are imported?
3. How many library functions are there?
4. Are there any conditional jumps?
5. Are there any unconditional jumps?
6. Are there any loops?
7. Which are the suspicious strings?
8. Is this a malicious file?
9. If yes, list out indicators.
10. How many variables are there?(Optional)

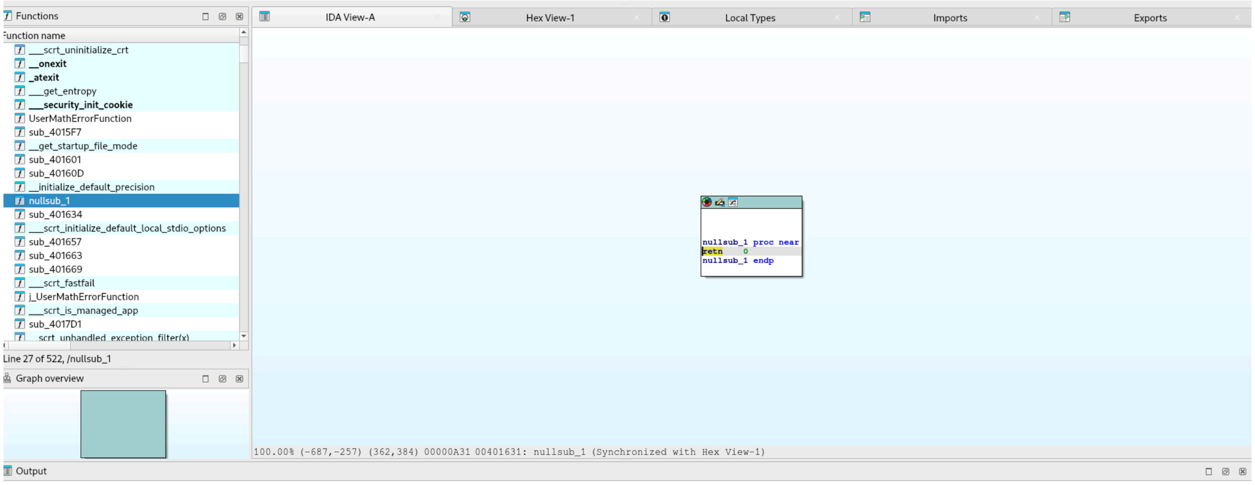
1. How many UDFs are there?



```
Function name
find_pe_section(uchar * const,uint)
___scrt_acquire_startup_lock
___scrt_initialize_crt
___scrt_initialize_onexit_tables
___scrt_is_nonwritable_in_current_image
___scrt_release_startup_lock
___scrt_uninitialize_crt
__onexit
_atexit
___get_entropy
___security_init_cookie
UserMathErrorFunction
sub_4015F7
__get_startup_file_mode
sub_401601
sub_40160D
__initialize_default_precision
nullsub_1
sub_401634
___scrt_initialize_default_local_stdio_options
sub_401657
sub_401663
Line 21 of 522, /UserMathErrorFunction
```

```
; int __cdecl UserMathErrorFunction(struct _exception *)
UserMathErrorFunction proc near
xor     eax, eax
retn
UserMathErrorFunction endp
```

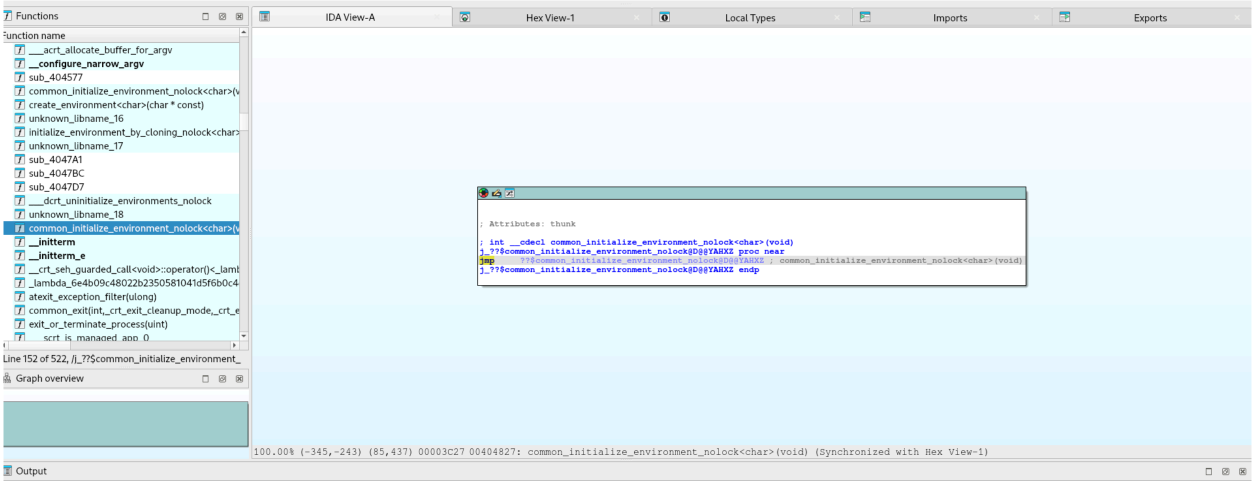
```
100.00% (-544,-243) (275,434) 000009F4 004015F4: UserMathErrorFunction (Synchronized with Hex View-1)
```



```
Function name
___scrt_uninitialize_crt
__onexit
_atexit
___get_entropy
___security_init_cookie
UserMathErrorFunction
sub_4015F7
__get_startup_file_mode
sub_401601
sub_40160D
__initialize_default_precision
nullsub_1
sub_401634
___scrt_initialize_default_local_stdio_options
sub_401657
sub_401663
sub_401669
___scrt_fastfail
j_UserMathErrorFunction
___scrt_is_managed_app
sub_4017D1
___scrt_unhandled_exception_filter(x)
Line 27 of 522, /nullsub_1
```

```
nullsub_1 proc near
retn    0
nullsub_1 endp
```

```
100.00% (-687,-257) (362,384) 00000A31 00401631: nullsub_1 (Synchronized with Hex View-1)
```



```
Function name
___acrt_allocate_buffer_for_argv
__configure_narrow_argv
sub_404577
common_initialize_environment_nolock<char>(v
create_environment<char>(char * const)
unknown_libname_16
initialize_environment_by_cloning_nolock<char>
unknown_libname_17
sub_4047A1
sub_4047BC
sub_4047D7
___dcrt_uninitialize_environments_nolock
unknown_libname_18
common_initialize_environment_nolock<char>(v
__initterm
__initterm_e
__crt_seh_guarded_call<void>::operator()<_lamb
_lambda_6e4b09c48022b2350581041d5f6b0c4
atexit_exception_filter(ulong)
common_exit(int,_crt_exit_cleanup_mode,_crt_e
exit_or_terminate_process(uint)
___scrt_is_managed_app_0
Line 152 of 522, /j_??$common_initialize_environment_
```

```
; Attributes: thunk

; int __cdecl common_initialize_environment_nolock<char>(void)
j_??$common_initialize_environment_nolock@D@@YAHXZ proc near
jmp     ??$common_initialize_environment_nolock@D@@YAHXZ ; common_initialize_environment_nolock<char>(void)
j_??$common_initialize_environment_nolock@D@@YAHXZ endp
```

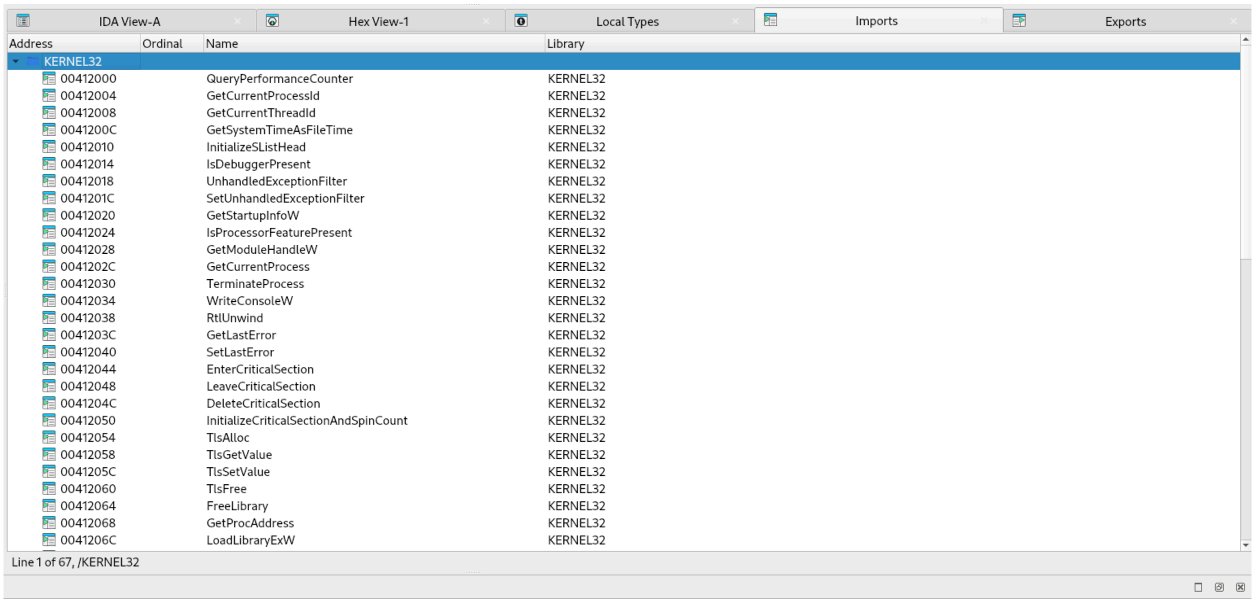
```
100.00% (-345,-243) (85,437) 00003C27 00404827: common_initialize_environment_nolock<char>(void) (Synchronized with Hex View-1)
```

One UDF was found namely UserMathError Function that xors the eax register value with itself. There is another nullsub_1 as well.

The common_initialize_environment can also be a UDF because it returns a cdecl which is a type of function call.

2. How many Libraries are imported?

The Kernel32 DLL is imported that further calls 66 imported libraries.



Address	Ordinal	Name	Library
KERNEL32			
00412000		QueryPerformanceCounter	KERNEL32
00412004		GetCurrentProcessId	KERNEL32
00412008		GetCurrentThreadId	KERNEL32
0041200C		GetSystemTimeAsFileTime	KERNEL32
00412010		InitializeSListHead	KERNEL32
00412014		IsDebuggerPresent	KERNEL32
00412018		UnhandledExceptionFilter	KERNEL32
0041201C		SetUnhandledExceptionFilter	KERNEL32
00412020		GetStartupInfoW	KERNEL32
00412024		IsProcessorFeaturePresent	KERNEL32
00412028		GetModuleHandleW	KERNEL32
0041202C		GetCurrentProcess	KERNEL32
00412030		TerminateProcess	KERNEL32
00412034		WriteConsoleW	KERNEL32
00412038		RtlUnwind	KERNEL32
0041203C		GetLastError	KERNEL32
00412040		SetLastError	KERNEL32
00412044		EnterCriticalSection	KERNEL32
00412048		LeaveCriticalSection	KERNEL32
0041204C		DeleteCriticalSection	KERNEL32
00412050		InitializeCriticalSectionAndSpinCount	KERNEL32
00412054		TlsAlloc	KERNEL32
00412058		TlsGetValue	KERNEL32
0041205C		TlsSetValue	KERNEL32
00412060		TlsFree	KERNEL32
00412064		FreeLibrary	KERNEL32
00412068		GetProcAddress	KERNEL32
0041206C		LoadLibraryExW	KERNEL32

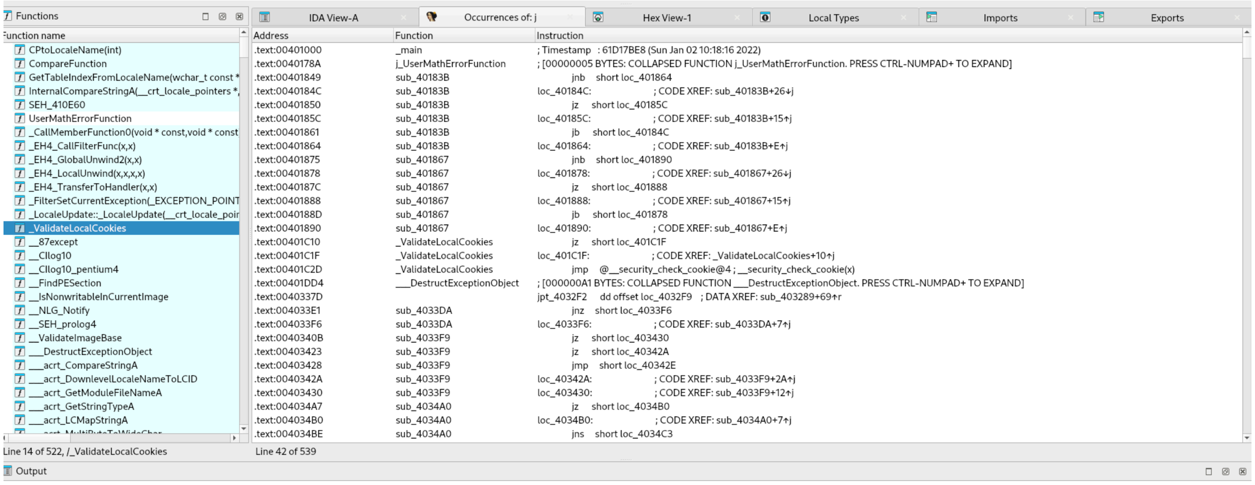
Line 1 of 67, /KERNEL32

3. How many library functions are there?

LIbrary functions are highlighted in “cyan” shade in the functions window.
66 library functions
They are also the ones imported by dll
Otherwise there are 449 library functions as seen in the functions window.

4. Are there any conditional jumps?

yes . occurrences of jnz (jump when not zero) and jz (jump when zero) indicate the presence of conditional jumps. They can also be seen through the red and green arrows in graph view.

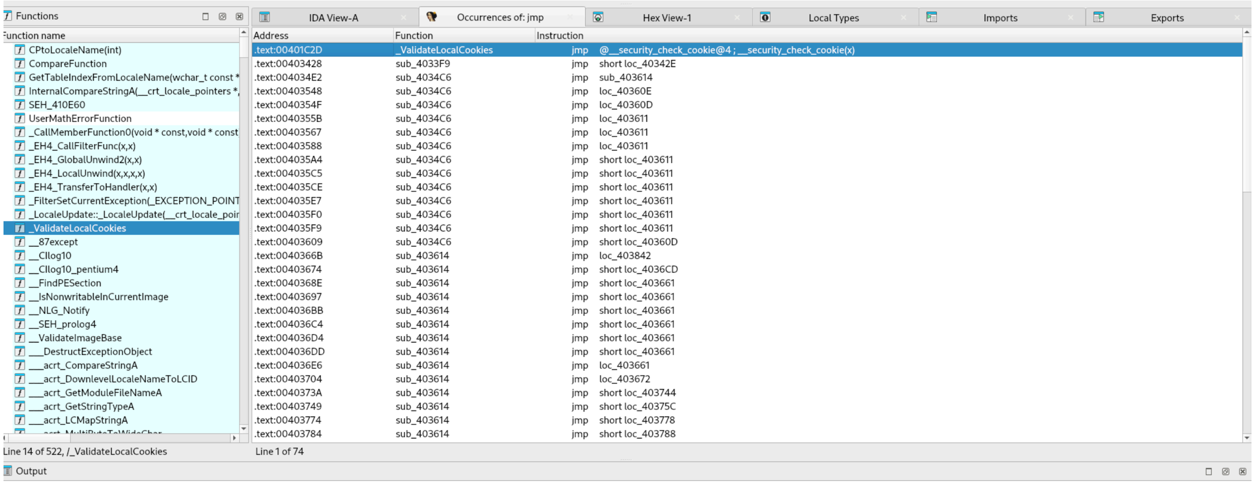


```
Function name
CPtoLocaleName(int)
CompareFunction
GetTableIndexFromLocaleName(wchar_t const *
InternalCompareStringA(__crt_locale_pointers *,
SEH_410E60
UserMathErrorFunction
_CallMemberFunction0(void * const,void * const
_EH4_CallFilterFunc(x,x)
_EH4_GlobalUnwind2(x,x)
_EH4_LocalUnwind(x,x,x,x)
_EH4_TransferToHandler(x,x)
_FilterSetCurrentException(_EXCEPTION_POINT
_LocaleUpdate::_LocaleUpdate(__crt_locale_poir
_ValidateLocalCookies
__87except
__CIlog10
__CIlog10_pentium4
__FindPESection
__IsNonwritableInCurrentImage
__NLG_Notify
__SEH_prolog4
__ValidateImageBase
___DestructExceptionObject
___acrt_CompareStringA
___acrt_DownlevelLocaleNameToLCID
___acrt_GetModuleFileNameA
___acrt_GetStringTypeA
___acrt_LCMapStringA
Line 14 of 522, /_ValidateLocalCookies
```

```
Address           Function                     Instruction
.text:00401000    _main                        ; Timestamp  : 61D17BE8 (Sun Jan 02 10:18:16 2022)
.text:0040178A    j_UserMathErrorFunction      ; [00000005 BYTES: COLLAPSED FUNCTION j_UserMathErrorFunction. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:00401849    sub_40183B                           jnb    short loc_401864
.text:0040184C    sub_40183B                   loc_40184C:                    ; CODE XREF: sub_40183B+26↓j
.text:00401850    sub_40183B                           jz     short loc_40185C
.text:0040185C    sub_40183B                   loc_40185C:                    ; CODE XREF: sub_40183B+15↑j
.text:00401861    sub_40183B                           jb     short loc_40184C
.text:00401864    sub_40183B                   loc_401864:                    ; CODE XREF: sub_40183B+E↑j
.text:00401875    sub_401867                           jnb    short loc_401890
.text:00401878    sub_401867                   loc_401878:                    ; CODE XREF: sub_401867+26↓j
.text:0040187C    sub_401867                           jz     short loc_401888
.text:00401888    sub_401867                   loc_401888:                    ; CODE XREF: sub_401867+15↑j
.text:0040188D    sub_401867                           jb     short loc_401878
.text:00401890    sub_401867                   loc_401890:                    ; CODE XREF: sub_401867+E↑j
.text:00401C10    _ValidateLocalCookies                jz     short loc_401C1F
.text:00401C1F    _ValidateLocalCookies        loc_401C1F:                    ; CODE XREF: _ValidateLocalCookies+10↑j
.text:00401C2D    _ValidateLocalCookies                jmp    @__security_check_cookie@4 ; __security_check_cookie(x)
.text:00401DD4    ___DestructExceptionObject   ; [000000A1 BYTES: COLLAPSED FUNCTION ___DestructExceptionObject. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:0040337D                                 jpt_4032F2    dd offset loc_4032F9   ; DATA XREF: sub_403289+69↑r
.text:004033E1    sub_4033DA                           jnz    short loc_4033F6
.text:004033F6    sub_4033DA                   loc_4033F6:                    ; CODE XREF: sub_4033DA+7↑j
.text:0040340B    sub_4033F9                           jz     short loc_403430
.text:00403423    sub_4033F9                           jz     short loc_40342A
.text:00403428    sub_4033F9                           jmp    short loc_40342E
.text:0040342A    sub_4033F9                   loc_40342A:                    ; CODE XREF: sub_4033F9+2A↑j
.text:00403430    sub_4033F9                   loc_403430:                    ; CODE XREF: sub_4033F9+12↑j
.text:004034A7    sub_4034A0                           jz     short loc_4034B0
.text:004034B0    sub_4034A0                   loc_4034B0:                    ; CODE XREF: sub_4034A0+7↑j
.text:004034BE    sub_4034A0                           jns    short loc_4034C3
Line 42 of 539
```

5. Are there any unconditional jumps?

Yes. There are 74 unconditional jumps. They can be seen via the jmp keywords.



```
Function name
CPtoLocaleName(int)
CompareFunction
GetTableIndexFromLocaleName(wchar_t const *
InternalCompareStringA(__crt_locale_pointers *,
SEH_410E60
UserMathErrorFunction
_CallMemberFunction0(void * const,void * const
_EH4_CallFilterFunc(x,x)
_EH4_GlobalUnwind2(x,x)
_EH4_LocalUnwind(x,x,x,x)
_EH4_TransferToHandler(x,x)
_FilterSetCurrentException(_EXCEPTION_POINT
_LocaleUpdate::_LocaleUpdate(__crt_locale_poir
_ValidateLocalCookies
__87except
__CIlog10
__CIlog10_pentium4
__FindPESection
__IsNonwritableInCurrentImage
__NLG_Notify
__SEH_prolog4
__ValidateImageBase
___DestructExceptionObject
___acrt_CompareStringA
___acrt_DownlevelLocaleNameToLCID
___acrt_GetModuleFileNameA
___acrt_GetStringTypeA
___acrt_LCMapStringA
Line 14 of 522, /_ValidateLocalCookies
```

```
Address           Function                 Instruction
.text:00401C2D    _ValidateLocalCookies        jmp   @__security_check_cookie@4 ; __security_check_cookie(x)
.text:00403428    sub_4033F9                   jmp   short loc_40342E
.text:004034E2    sub_4034C6                   jmp   sub_403614
.text:00403548    sub_4034C6                   jmp   loc_40360E
.text:0040354F    sub_4034C6                   jmp   loc_40360D
.text:0040355B    sub_4034C6                   jmp   loc_403611
.text:00403567    sub_4034C6                   jmp   loc_403611
.text:00403588    sub_4034C6                   jmp   loc_403611
.text:004035A4    sub_4034C6                   jmp   short loc_403611
.text:004035C5    sub_4034C6                   jmp   short loc_403611
.text:004035CE    sub_4034C6                   jmp   short loc_403611
.text:004035E7    sub_4034C6                   jmp   short loc_403611
.text:004035F0    sub_4034C6                   jmp   short loc_403611
.text:004035F9    sub_4034C6                   jmp   short loc_403611
.text:00403609    sub_4034C6                   jmp   short loc_40360D
.text:0040366B    sub_403614                   jmp   loc_403842
.text:00403674    sub_403614                   jmp   short loc_4036CD
.text:0040368E    sub_403614                   jmp   short loc_403661
.text:00403697    sub_403614                   jmp   short loc_403661
.text:004036BB    sub_403614                   jmp   short loc_403661
.text:004036C4    sub_403614                   jmp   short loc_403661
.text:004036D4    sub_403614                   jmp   short loc_403661
.text:004036DD    sub_403614                   jmp   short loc_403661
.text:004036E6    sub_403614                   jmp   loc_403661
.text:00403704    sub_403614                   jmp   loc_403672
.text:0040373A    sub_403614                   jmp   short loc_403744
.text:00403749    sub_403614                   jmp   short loc_40375C
.text:00403774    sub_403614                   jmp   short loc_403778
.text:00403784    sub_403614                   jmp   short loc_403788
Line 1 of 74
```

6. Are there any loops?

No.

7. Which are the suspicious strings?

```
.rdata:00412B0C      00000016      C       `anonymous namespace'
.rdata:004184B0      00000012      C       GetCurrentProcess
.rdata:00412C20      0000001C      C       InitializeCriticalSectionEx
.rdata:0041380C      00000025      C       AppPolicyGetProcessTerminationMethod
.rdata:00418420      00000012      C       IsDebuggerPresent
.rdata:0041846E      00000010      C       GetStartupInfoW
.rdata:004184C4      00000011      C       TerminateProcess
.rdata:00418512      00000015      C       EnterCriticalSection
.rdata:0041852A      00000015      C       LeaveCriticalSection
.rdata:00418542      00000016      C       DeleteCriticalSection
.rdata:004185D4      0000000F      C       LoadLibraryExW
```

8. Is this a malicious file?

Yes. This can be concluded from aforementioned suspicious strings.

9. If yes, list out indicators.

Suspicious Strings
Some Imported library functions like GetProcAddress and LoadLibrary

10. How many variables are there?(Optional)

There are 7 static variables.

```
.text:00401000 var_C           = dword ptr -0Ch
.text:00401000 var_8           = dword ptr -8
.text:00401000 var_4           = dword ptr -4
xt:00401050 arg_0              = dword ptr  8
.text:00401050 arg_4           = dword ptr  0Ch
.text:00401050 arg_8           = dword ptr  10h
.text:00401050 arg_C           = dword ptr  14h
```