

# Automatically detect and support against anti-debug with IDA/Ghidra to streamline debugging process

---

Takahiro Takeda  
Cyber Emergency Center, LAC



# Profile

## Takahiro Takeda

Malware Analysis Team



2016: Analysis work as a Security Analyst.

2017: Analyzing malware and logs, as well as investigating smishing.

2019: Mainly Responsible for malware analysis related to incidents.

Speaker Experience:

PACSEC, AVAR, HITCON, Black Hat USA Arsenal, Virus Bulletin

# Agenda

- What is AntiDebugSeeker
- The motivation behind developing this tool
- Demo  
**IDA version**
- Introduction to Configuration Files
- Demo  
**GHIDRA version**
- Introduction to Files related to Ghidra Version
- AntiDebugSeeker VS .Net Base Malware
- Summary

This is a program for automatically identify and extract potential anti-debugging techniques used by malware and displaying them in **IDA / Ghidra**.

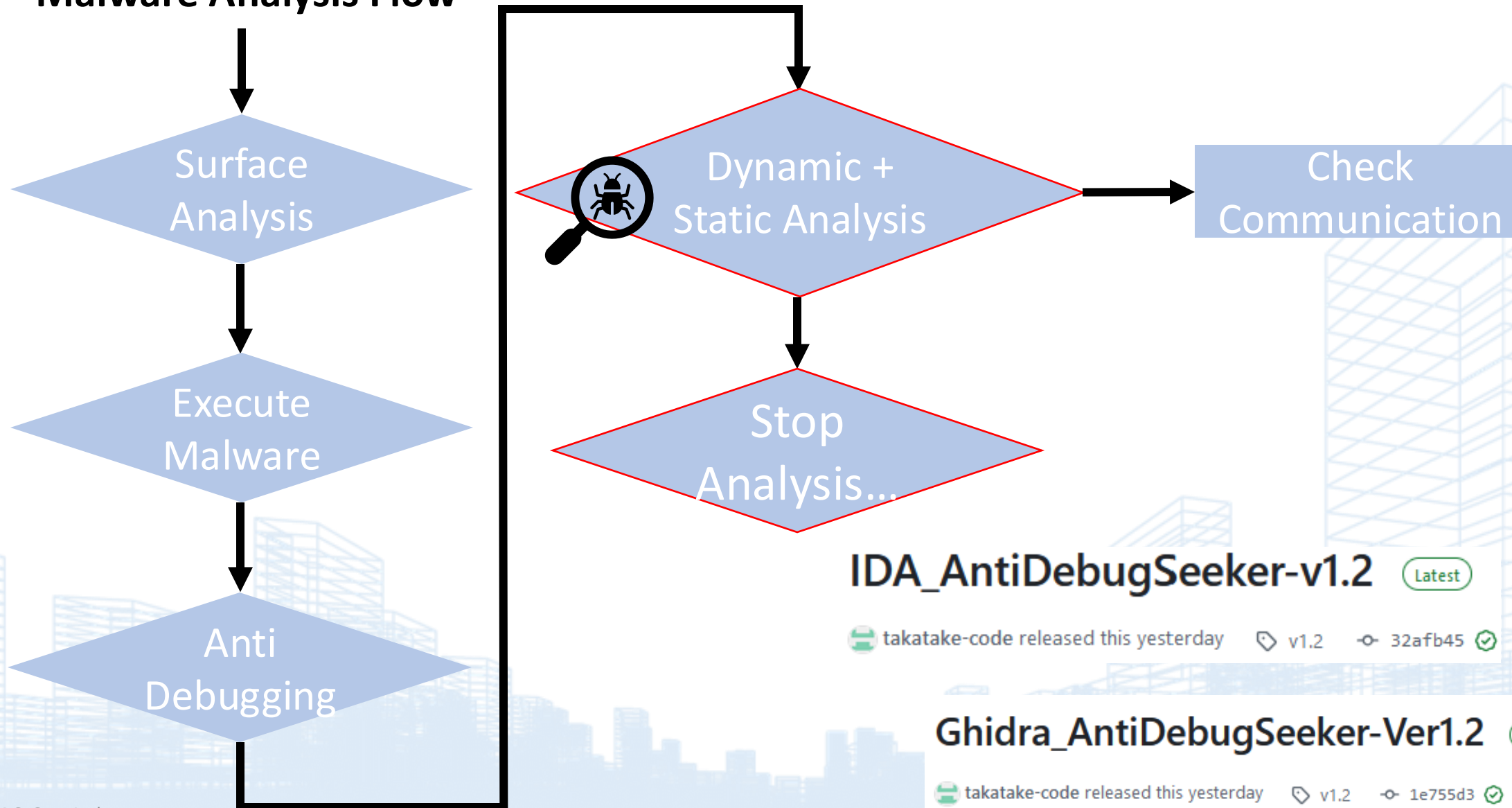
The main functionalities of this plugin are as follows:

1. Extraction of APIs that are potentially being used for anti-debugging by the malware.
2. Using multiple keywords, anti-debugging techniques are extracted.

※ For packed samples, running this plugin after unpacking and fixing the Import Address Table is more effective.

# The motivation behind developing this tool

## Malware Analysis Flow



IDA\_AntiDebugSeeker-v1.2 Latest

takatake-code released this yesterday v1.2 32afb45

Ghidra\_AntiDebugSeeker-Ver1.2 Latest

takatake-code released this yesterday v1.2 1e755d3

# Demo: IDA version of AntiDebugSeeker

Malware : Ursnif

MD5 : 4da11c829f8fea1b690f317837af8387 (Packed)

MD5 : 952d604345e051fce76729ccb63bde82 (Unpack)

The flow of a demo

- ① A type of anti-analysis leads to the termination of the process.
- ② Using AntiDebugSeeker to find anti-analysis features.
- ③ Apply patches using a debugger.

Process Hacker [DESKTOP-CJ7SNMKWin10] - (Administrator)

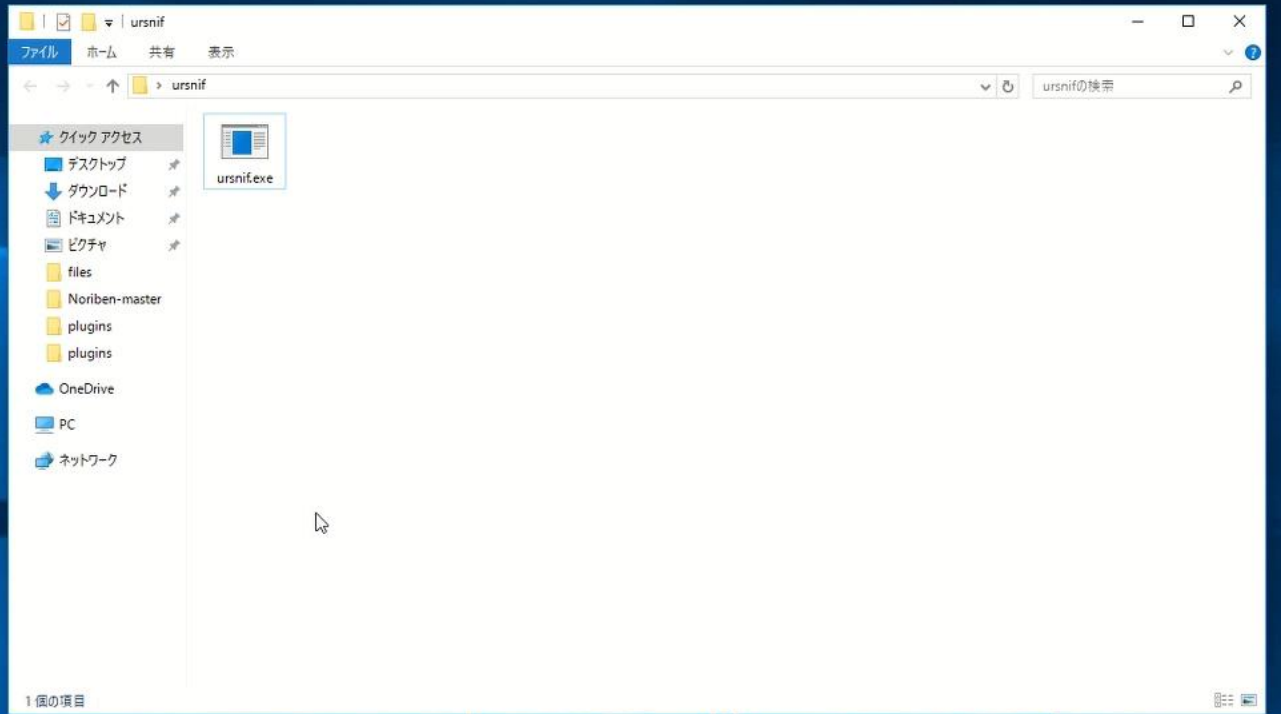
Hacker View Tools Users Help

Refresh Options Find handles or DLLs System information Search Processes (Ctrl+K)

Processes Services Network Disk

Name	PID	CPU	I/O tot...	Private...	User name	Description
svchost.exe	80			8.97 MB	...%LOCAL SERVICE	Windows サービスのホス...
svchost.exe	272	0.18	2.16 k...	13.44 ...	...%LOCAL SERVICE	Windows サービスのホス...
svchost.exe	684		88 B/s	12.52 ...	NT AUT...%SYSTEM	Windows サービスのホス...
svchost.exe	1160	0.02	568 B/s	6.36 MB	...%NETWORK SER\	Windows サービスのホス...
svchost.exe	1332			2.32 MB	...%LOCAL SERVICE	Windows サービスのホス...
svchost.exe	1388			1.88 MB	...%LOCAL SERVICE	Windows サービスのホス...
spoolsv.exe	1508			5.4 MB	NT AUT...%SYSTEM	スプラー サブシステム アプ...
svchost.exe	1864			9.19 MB	NT AUT...%SYSTEM	Windows サービスのホス...
svchost.exe	1892			6.26 MB	NT AUT...%SYSTEM	Windows サービスのホス...
vmtoolsd.exe	1900	0.09	965 B/s	6.67 MB	NT AUT...%SYSTEM	VMware Tools Core Se...
vm3dservice.exe	1908			1.4 MB	NT AUT...%SYSTEM	VMware SVGA Helper ...
vm3dservice...	2084			1.52 MB	NT AUT...%SYSTEM	VMware SVGA Helper ...
VGAAuthService...	1920			2.65 MB	NT AUT...%SYSTEM	VMware Guest Authen...
dllhost.exe	2372			3.77 MB	NT AUT...%SYSTEM	COM Surrogate
msdtc.exe	2652			2.46 MB	...%NETWORK SER\	Microsoft 分散トランザ...
svchost.exe	512			1.74 MB	...%LOCAL SERVICE	Windows サービスのホス...
SearchIndexer.e...	348			28.05 ...	NT AUT...%SYSTEM	Microsoft Windows Se...
svchost.exe	1364			6.43 MB	DESKTO...%Win10	Windows サービスのホス...
svchost.exe	3032			1.59 MB	NT AUT...%SYSTEM	Windows サービスのホス...
svchost.exe	6560			1.53 MB	NT AUT...%SYSTEM	Windows サービスのホス...
lsass.exe	624	0.08		4.67 MB	NT AUT...%SYSTEM	Local Security Authorit...
csrss.exe	492	0.04		1.91 MB	NT AUT...%SYSTEM	クライアント サーバー ランタ...
winlogon.exe	564			3.65 MB	NT AUT...%SYSTEM	Windows ログオン アプリ...
dwm.exe	884	0.07		111.38...	Windo...%DWM-1	デスクトップ ウィンドウ マネ...
explorer.exe	3292	0.11		302.07...	DESKTO...%Win10	エクスプローラー
MSASCuiL.exe	1680			2.81 MB	DESKTO...%Win10	Windows Defender no...
vmtoolsd.exe	124	0.07	684 B/s	20.14 ...	DESKTO...%Win10	VMware Tools Core Se...
OneDrive.exe	6288			16.94 ...	DESKTO...%Win10	Microsoft OneDrive
ProcessHacker.exe	1468	0.53		17.59 ...	DESKTO...%Win10	Process Hacker
sakura.exe	7016			3.89 MB	DESKTO...%Win10	サクラエディタ

CPU Usage: 4.36% Physical memory: 1.28 GB (31.98%) Processes: 52





- ごみ箱
- x32dbg.exe - ショートカット
- files
- x64dbg.exe - ショートカット
- ursnif
- サクラエディタ
- Cywin64 Terminal
- desktop.ini
- Firefox
- desktop.ini
- DA Pro 8.3 (32-bit)
- DA Pro 8.3 (64-bit)
- dnSpy.exe - ショートカット
- ProcessHacker.exe - ショートカット
- procepx64 - ショートカット
- Procmon - ショートカット

Application window titled "ursnif" showing a file explorer view of the "ursnif" directory. The window title bar includes "アプリケーション ツール" and "ursnif". The address bar shows the path "ursnif". The left sidebar lists "クイック アクセス" (Quick Access) with items: デスクトップ (Desktop), ダウンロード (Downloads), ドキュメント (Documents), ピクチャ (Pictures), files, Nonben-master, plugins, plugins, OneDrive, PC, and ネットワーク (Network). The main pane displays a single file named "ursnif.exe". The status bar at the bottom indicates "1 個の項目" (1 item) and "1 個の項目を選択 277 KB" (1 item selected 277 KB).

# The Analysis result of IDA-AntiDebugSeeker

Detected Function List

ti Debug Detection Resu

Search...

sub\_401000  
(0x401000)

- SetupDiEnumDeviceInfo
- SetupDiGetClassDevsA
- SetupDiGetDeviceRegistryPropertyA
- SetupDiGetDeviceRegistryPropertyA
- SetupDiGetDeviceRegistryPropertyA

(5detected)

sub\_401395  
(0x401395)

- GetCursorInfo
- CloseHandle
- CloseHandle
- CloseHandle
- CloseHandle
- Opened\_Exclusively\_Check

(7detected)

It was determined that the function sub\_401000 has anti-debugging features.

Detected Function List

ti Debug Detection Resu

Hex View-1

Category Name	Possible Anti-Debug API	Address
Analysis Environment Check	SetupDiGetClassDevsA	0x401022
Analysis Environment Check	SetupDiEnumDeviceInfo	0x401043
Analysis Environment Check	SetupDiGetDeviceRegistryPr...	0x401062
Analysis Environment Check	SetupDiGetDeviceRegistryPr...	0x401068
Analysis Environment Check	SetupDiGetDeviceRegistryPr...	0x401092
Check Invalid Close->Exception	CloseHandle	0x401410
Check Invalid Close->Exception	CloseHandle	0x401419
Check Invalid Close->Exception	CloseHandle	0x40141E
User Interaction Check	GetCursorInfo	0x40161B
Check Invalid Close->Exception	CloseHandle	0x401707
Time Check	Sleep	0x40184F
Check Invalid Close->Exception	CloseHandle	0x40185D
Check Invalid Close->Exception	CloseHandle	0x40194D
Time Check	Sleep	0x4019A8
Memory Manipulation	VirtualProtectEx	0x4019C7
Memory Manipulation	VirtualProtectEx	0x4019DD
Memory Manipulation	VirtualProtectEx	0x401A11
Check Invalid Close->Exception	CloseHandle	0x401E35
Thread Execute	ResumeThread	0x402170
Time Check	WaitForSingleObject	0x40217E
Thread Manipulation	CloseHandle	0x402191
Thread Execute		
Time Check		
Thread Manipulation		
Thread Execute		
Check Invalid Close		
Check Invalid Close		

It also informs us about aspects related to malware functions, such as memory manipulation.

# Comment Function

```
push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF8h
mov     eax, large fs:30h ; NtGlobalFlag_check - The code is checking the NtGlobalFlag value at offset 0x68 from the Process Environment Block.
                                ; The value 70 is the sum of FLG_HEAP_ENABLE_TAIL_CHECK (0x10), FLG_HEAP_ENABLE_FREE_CHECK (0x20), and FLG_HEAP_VALIDATE_PARAMETERS (0x40).
sub     esp, 480h
test    byte ptr [eax+68h], 70h
push    esi
push    edi
jz      short loc_4BFFB2

; .text:00402170
push    04h ; dwMilliseconds
push    0 ; dwFlagsAndAttributes
push    3 ; dwCreationDisposition
push    edi ; lpSecurityAttributes
push    1 ; dwShareMode
push    80000000h ; dwDesiredAccess | Opened_Exclusively_Check - CreateFile is attempting to exclusively open its own executable file.
                                ; If it fails to do so, it deduces that a debugger may already have it open exclusively. If the dwShareMode argument of CreateFile is 0, this is highly likely.
or      ebx, 0FFFFFFFh
push    eax ; lpFileName
mov     [ebp+CreationTime.dwLowDateTime], ebx
mov     [ebp+CreationTime.dwHighDateTime], ebx
call    ds:CreateFileA
mov     [ebp+hObject], eax

; .text:0040218E
push    dword ptr [edi+4] ; hThread

push    ebp
mov     ebp, esp
push    ecx
mov     eax, large fs:30h ; BeingDebugged_check - The BeingDebugged field in the Process Environment Block (PEB) indicates whether the current process is being debugged or not.
movzx   cax, byte ptr [cax+2]
test    eax, eax
setnz   byte ptr [ebp+var_4]
cmp     [ebp+var_4], 0
jz      short loc_40102E
```

# Extra Function - Edit Config File -

The screenshot displays the IDA Pro interface with the following assembly code blocks:

```
.text:00401706 loc_401706:           ; hObject
.text:00401706 push  esi
.text:00401707 call  ds:CloseHandle ; Check Invalid Close->Exception

.text:0040170D loc_40170D:
.text:0040170D cmp    [esp+78h+lpszShortPath], ebx
.text:00401711 jz     short loc_401724

.text:00401713 call  sub_401000
.text:00401718 cmp    eax, ebx
.text:0040171A jz     short loc_401724

.text:00401724 loc_401724:
.text:00401724 call  sub_40144C
.text:00401729 mov    ecx, dword_407664
.text:0040172F xor    ecx, eax
.text:00401731 push  ecx           ; ResultLength
.text:00401732 mov    dword_4075D4, eax
.text:00401737 call  sub_402A9B
.text:0040173C mov    dword_407628, eax
.text:00401741 cmp    eax, ebx
.text:00401743 jnz    short loc_40175D
```

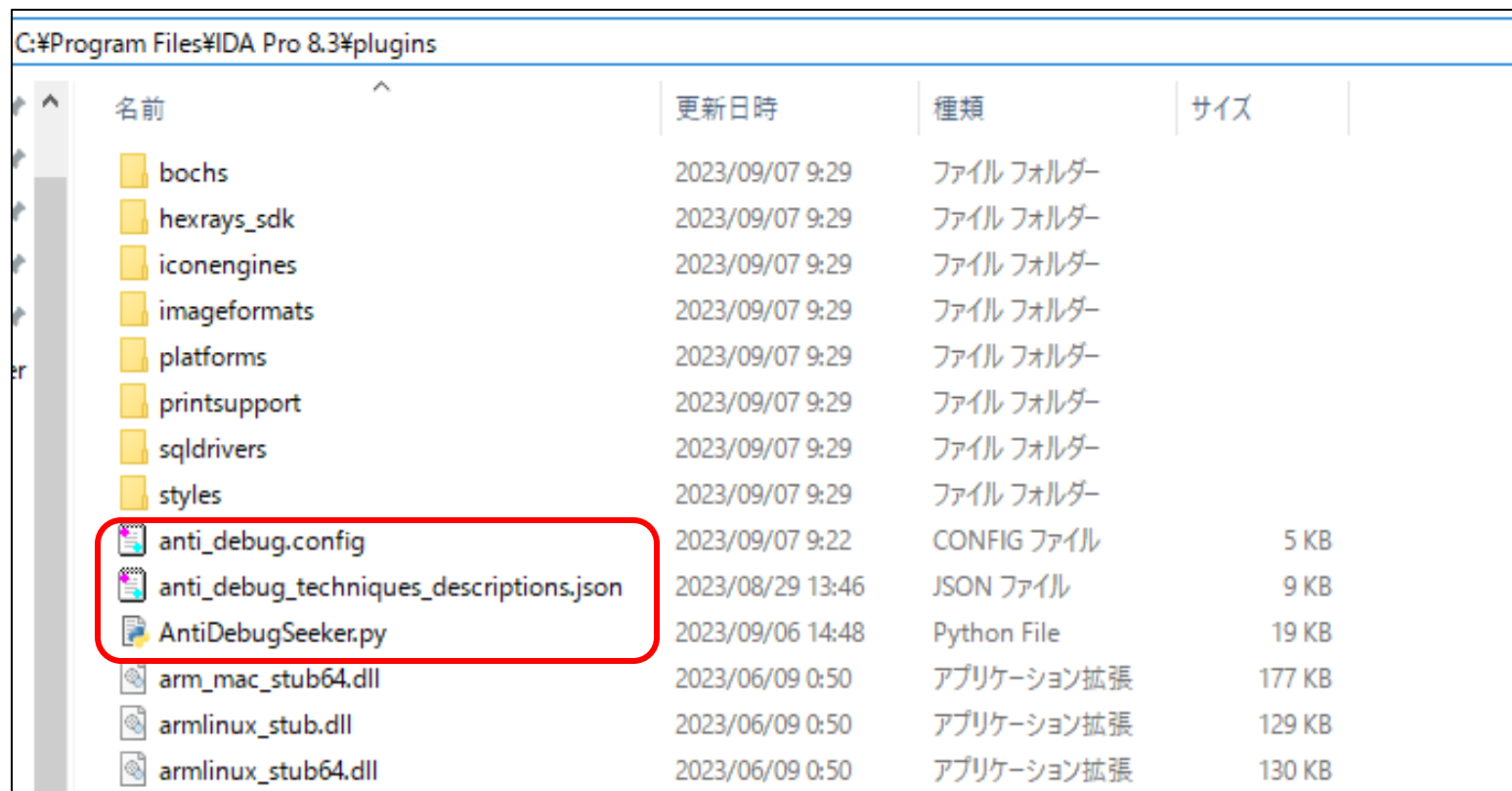
The status bar at the bottom indicates: 100.00% (7,4337) (420,288) 00000813 00401713: sub\_401571+1A2 (Synchronized with Hex View-1)

The Output window at the bottom shows the following messages:

```
Nothing Found for pattern NtQuerySystemInformation_KD_Check.
Nothing Found for pattern Extract_Resource_Section.
Nothing Found for pattern Commucate_function_String.
Nothing Found for pattern Commucate_function.
AntiDebugSeeker terminated.
Edit anti_debug.config : Switch Other tab and Press Ctrl+Shift+E.
Checking the recursive calls : sub_401000
```

# Introduction to configuration files

# Files Required to Run the Program



名前	更新日時	種類	サイズ
bochs	2023/09/07 9:29	ファイル フォルダー	
hexrays_sdk	2023/09/07 9:29	ファイル フォルダー	
iconengines	2023/09/07 9:29	ファイル フォルダー	
imageformats	2023/09/07 9:29	ファイル フォルダー	
platforms	2023/09/07 9:29	ファイル フォルダー	
printsupport	2023/09/07 9:29	ファイル フォルダー	
sqldrivers	2023/09/07 9:29	ファイル フォルダー	
styles	2023/09/07 9:29	ファイル フォルダー	
anti_debug.config	2023/09/07 9:22	CONFIG ファイル	5 KB
anti_debug_techniques_descriptions.json	2023/08/29 13:46	JSON ファイル	9 KB
AntiDebugSeeker.py	2023/09/06 14:48	Python File	19 KB
arm_mac_stub64.dll	2023/06/09 0:50	アプリケーション拡張	177 KB
armlinux_stub.dll	2023/06/09 0:50	アプリケーション拡張	129 KB
armlinux_stub64.dll	2023/06/09 0:50	アプリケーション拡張	130 KB

Please place the following three files under the plugin directory of IDA :

- 1.anti\_debug.config (A file containing rules for detecting anti-debugging techniques)
- 2.anti\_debug\_techniques\_descriptions.json (A file containing descriptions of the detected rules)
- 3.AntiDebugSeeker.py (The anti-debugging detection program)

## Anti\_Debug\_API

```
###Anti_Debug_API###
```

```
[CommandLine check]
```

```
GetCommandLineA
```

```
GetCommandLineW
```

```
[Debugger check]
```

```
CheckRemoteDebuggerPresent
```

```
DebugActiveProcess
```

```
DebugBreak
```

```
DbgSetDebugFilterState
```

```
DbgUiDebugActiveProcess
```

```
IsDebuggerPresent
```

```
NtDebugActiveProcess
```

```
NtQueryObject
```

```
NtSetDebugFilterState
```

```
NtSystemDebugControl
```

```
OutputDebugStringA
```

```
OutputDebugStringW
```

In the Anti\_Debug\_API section, you can freely create categories and add any number of APIs you want to detect.  
(**exact match**)

```
###Anti_Debug_API###
```

```
[Category Name_1]
```

```
API1
```

```
API2
```

```
API3
```

```
[Category Name_2]
```

```
API4
```

```
API5
```

```
API6
```

## Anti\_Debug\_Technique

```
###Anti_Debug_Technique###  
default_search_range=80
```

```
[VMware_I/O_port]  
5658h
```

```
[VMware_magic_value]  
564D5868h
```

```
[HeapTailMarker]  
ABABABAB
```

```
[KernelDebuggerMarker]  
7FFE02D4
```

```
[DbgBreakPoint_RET]  
DbgBreakPoint  
C3h
```

```
[DbgUiRemoteBreakin_Debugger_Terminate]  
DbgUiRemoteBreakin  
TerminateProcess
```

You can set up to three keywords (partial match) under a single rule name.

```
###Anti_Debug_Technique###  
default_search_range=80
```

```
[Rule1]
```

```
ABC ] 80bytes
```

```
DEF ] 80bytes
```

```
GHI
```

```
search_range=200
```

Search Target:

Disassembly (Opcode, Operand)

Comments

API based on Import Table



```
1 {
2   "VMware_I/O_port" : "detect a VM environment based on the VMware I/O port",
3   "VMware_magic_value" : "detect a VM environment based on the VMware magic value",
4   "HeapTailMarker": "Malware can detect if it's on a debug heap by checking the heap tail marker",
5   "KernelDebuggerMarker": "Detect Kernelmode Debugger(KdDebuggerEnabled)",
6   "DbgBreakPoint_RET": "This detection may be due to the first byte of the DbgBreakPoint RET instruction",
7   "DbgUiRemoteBreakin_Debugger_Terminate": "When a debugger tries to attach to a process, it sends a DbgUiRemoteBreakin message to the process. The debugger terminates the process if the process is being debugged.",
8   "PMCCheck_RDPMC": "The RDPMC (Read Performance-Monitoring Counters) instruction can be used to detect if a process is running in a VM.",
9   "TimingCheck_RDTSC": "The RDTSC (Read Time Stamp Counter) instruction can be used to detect if a process is running in a VM.",
10  "Environment_TimingCheck_CPUID": "The CPUID instruction can be used as part of an anti-debugging technique.",
11  "SkipPrefixes_INT1": "This anti-debugging method exploits how some debuggers handle the INT1 instruction.",
12  "INT2D_interrupt_check": "The INT2D instruction either passes control to a debugger or the process.",
13  "INT3_interrupt_check": "This is a debug detection mechanism using the INT3 instruction.",
14  "EXCEPTION_BREAKPOINT": "This is a debug detection method using the INT3 instruction.",
15  "ICE_interrupt_check": "If a program is debugged, the debugger sees the exception.",
16  "DBG_PRINTEXCEPTION_C": "This may involve anti-debugging by utilizing the DBG_PRINTEXCEPTION_C instruction.",
17  "TrapFlag_SingleStepException": "This anti-debugging technique utilizes the TrapFlag and SingleStepException.",
18  "BeingDebugged_check" : "The BeingDebugged field in the Process Environment Block (PEB) is used to detect if a process is being debugged.",
19  "NtGlobalFlag_check": "The code is checking the NtGlobalFlag value at offset 0x00000000.",
20  "NtGlobalFlag_check_2": "The code is checking the NtGlobalFlag value at offset 0x00000001.",
21  "HeapFlags" : "HeapFlags stores various heap-related flags, bit by bit. \nThese flags are used to detect if a process is running in a VM."
}
```

## Anti\_Debug\_Technique

```
###Anti_Debug_Technique###
default_search_range=80
```

```
[VMware_I/O_port]
5658h
```

```
[VMware_magic_value]
564D5868h
```

```
[HeapTailMarker]
ABABABAB
```

```
[KernelDebuggerMarker]
7FFE02D4
```

```
[DbgBreakPoint_RET]
DbgBreakPoint
C3h
```

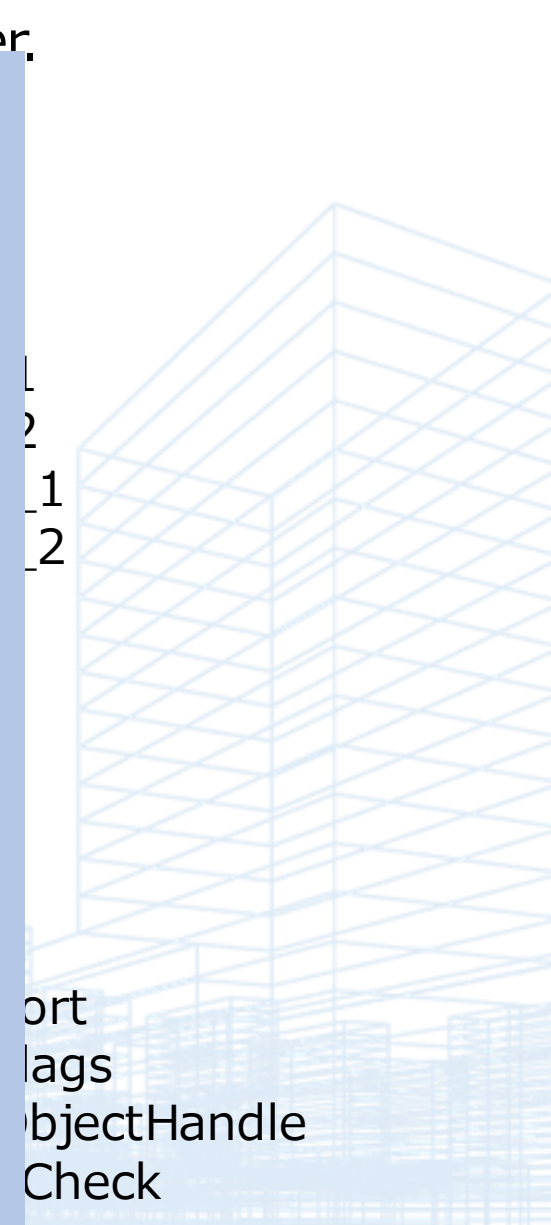
```
[DbgUiRemoteBreakin_Debugger_Terminate]
DbgUiRemoteBreakin
TerminateProcess
```

```
ABABABAB, i
C3, which co
inates.",
MC) to deter
utilized in
volves check
ion prefixes
value if no d
if the progr
e program is
o bit in the
triggered by
ecimal 100)
process is b
\nThe value
t Block. \nT
in features
```

# List of detectable anti-debugging techniques (Ver1.0)

The following Anti Debug Techniques can be detected using AntiDebugSeeker.

HeapTailMarker	VM_Check
KernelDebugg	VBox_Check
DbgBreakPoint	VMware_Check
DbgUiRemote	VMware_I/O_port
PMCCheck_RD	VMware_magic_value
TimingCheck_	CreateMutex_AlreadyExist
SkipPrefixes_I	CreateEvent_AlreadyExist
INT2D_interru	ChildProcess_Check
INT3_interrup	Extract_Resource_Section
EXCEPTION_B	Commucate_function_String
ICE_interrupt_	Commucate_function
DBG_PRINTEX	NtSetInformationThread
TrapFlag_Sing	NtQueryInformationProcess
BeingDebugge	Anti-Sandbox_SandBoxie
NtGlobalFlag_	Anti-Sandbox_Buster_Sandbox_Analyzer
NtGlobalFlag_	
HeapFlags	
HeapForceFlag	
Combination_	
Combination_of_HEAP_Flags_2	



ort  
lags  
bjectHandle  
Check

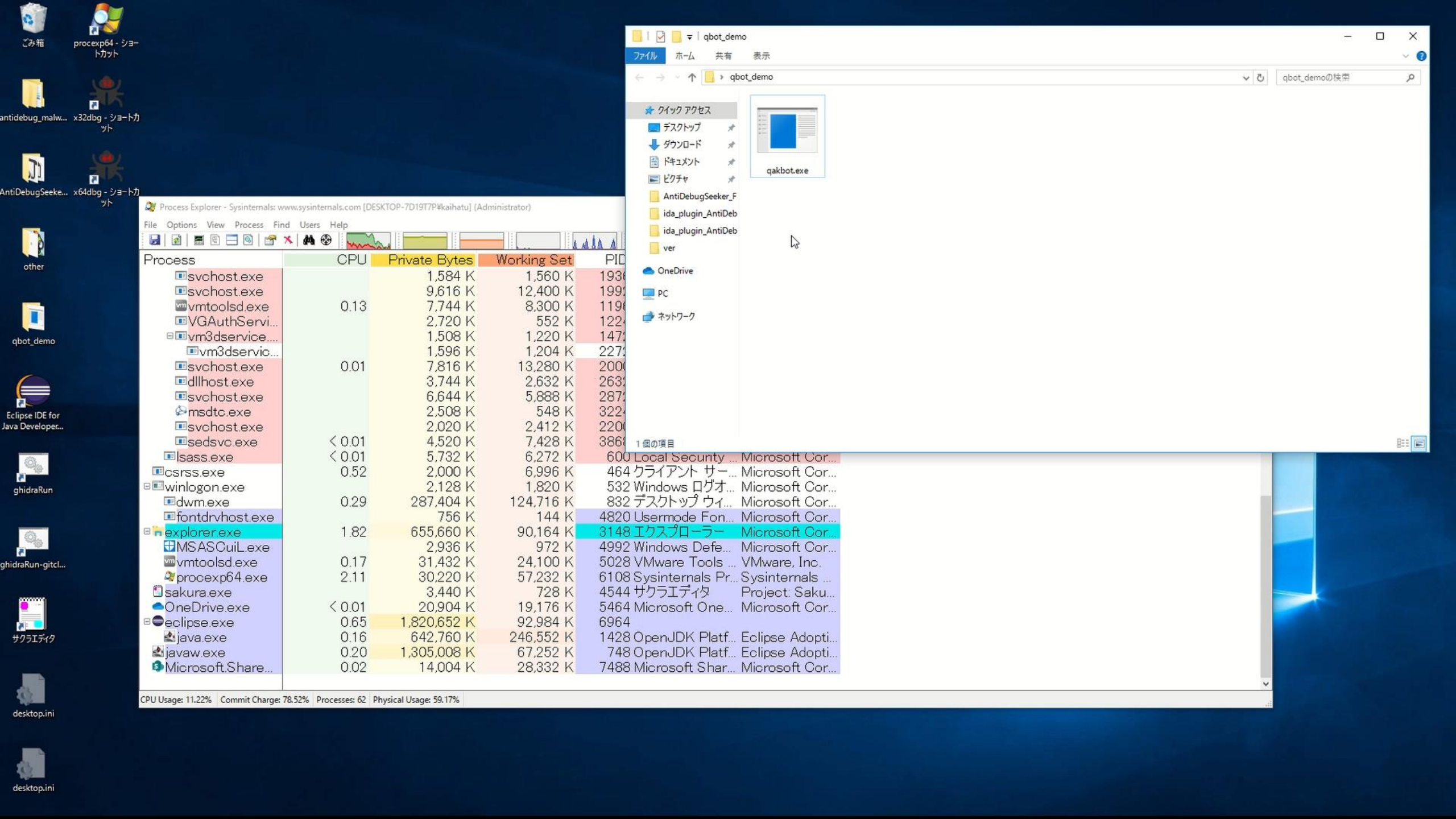
# Demo: Ghidra version of AntiDebugSeeker

Malware : Qakbot (aka. Qbot)

- ❑ MD5 : bce0df8721504d50f4497c0a0a2c090d (Packed)
- ❑ MD5 : 58e1c32eeb0130da19625e55ee48cf1e (Unpack)

The flow of a demo

- ① A type of anti-analysis leads to the termination of the process.
- ② Using AntiDebugSeeker to find anti-analysis features.
- ③ Examine the behavior of AntiDebug, and identify the areas to patch from the AntiDebugSeeker results + Apply the patch using a debugger.



Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-7D19T7P%kaihatu] (Administrator)

Process	CPU	Private Bytes	Working Set	PID
svchost.exe		1,584 K	1,560 K	1936
svchost.exe		9,616 K	12,400 K	1992
vmtoolsd.exe	0.13	7,744 K	8,300 K	1192
VGAuthService.exe		2,720 K	552 K	1224
vm3dservice.exe		1,508 K	1,220 K	1472
vm3dservice.exe		1,596 K	1,204 K	2272
svchost.exe	0.01	7,816 K	13,280 K	2000
dllhost.exe		3,744 K	2,632 K	2632
svchost.exe		6,644 K	5,888 K	2872
msdtc.exe		2,508 K	548 K	3224
svchost.exe		2,020 K	2,412 K	2200
sedsvc.exe	< 0.01	4,520 K	7,428 K	3868
lsass.exe	< 0.01	5,732 K	6,272 K	600
csrss.exe	0.52	2,000 K	6,996 K	464
winlogon.exe		2,128 K	1,820 K	532
dwm.exe	0.29	287,404 K	124,716 K	832
fontdrvhost.exe		756 K	144 K	4820
explorer.exe	1.82	655,660 K	90,164 K	3148
MSASCuiL.exe		2,936 K	972 K	4992
vmtoolsd.exe	0.17	31,432 K	24,100 K	5028
procexp64.exe	2.11	30,220 K	57,232 K	6108
sakura.exe		3,440 K	728 K	4544
OneDrive.exe	< 0.01	20,904 K	19,176 K	5464
eclipse.exe	0.65	1,820,652 K	92,984 K	6964
java.exe	0.16	642,760 K	246,552 K	1428
javaw.exe	0.20	1,305,008 K	67,252 K	748
Microsoft.Share...	0.02	14,004 K	28,332 K	7488

1 個の項目

600	Local Security ...	Microsoft Cor...
464	クライアント サー...	Microsoft Cor...
532	Windows ログオ...	Microsoft Cor...
832	デスクトップ ウィ...	Microsoft Cor...
4820	Usermode Fon...	Microsoft Cor...
3148	エクスプローラー	Microsoft Cor...
4992	Windows Defe...	Microsoft Cor...
5028	VMware Tools ...	VMware, Inc.
6108	Sysinternals Pr...	Sysinternals ...
4544	サクラエディタ	Project: Saku...
5464	Microsoft One...	Microsoft Cor...
6964		
1428	OpenJDK Platf...	Eclipse Adopti...
748	OpenJDK Platf...	Eclipse Adopti...
7488	Microsoft Shar...	Microsoft Cor...

CPU Usage: 11.22% Commit Charge: 78.52% Processes: 62 Physical Usage: 59.17%

- ごみ箱
- procexp64 - ショートカット
- antidebug\_malw... x32dbg - ショートカット
- AntiDebugSeeker... x64dbg - ショートカット
- other
- qbot\_demo
- Eclipse IDE for Java Developer...
- ghidraRun
- ghidraRun-gitcl...
- サクラエディタ
- desktop.ini
- desktop.ini

Ghidra: Demo

File Edit Project Tools Help

Tool Chest

Active Project: Demo

- Demo
  - qakbot.exe

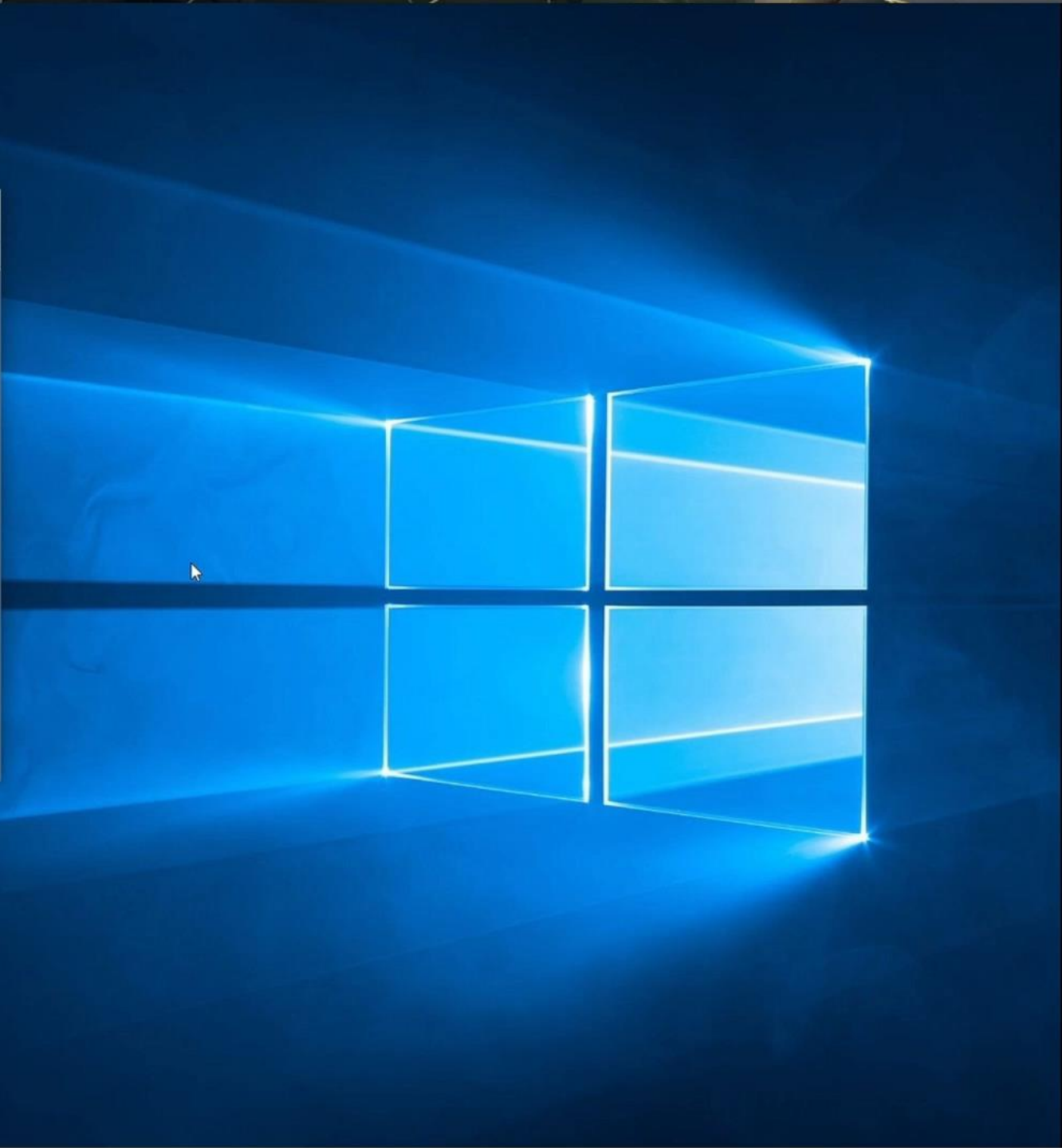
Filter:

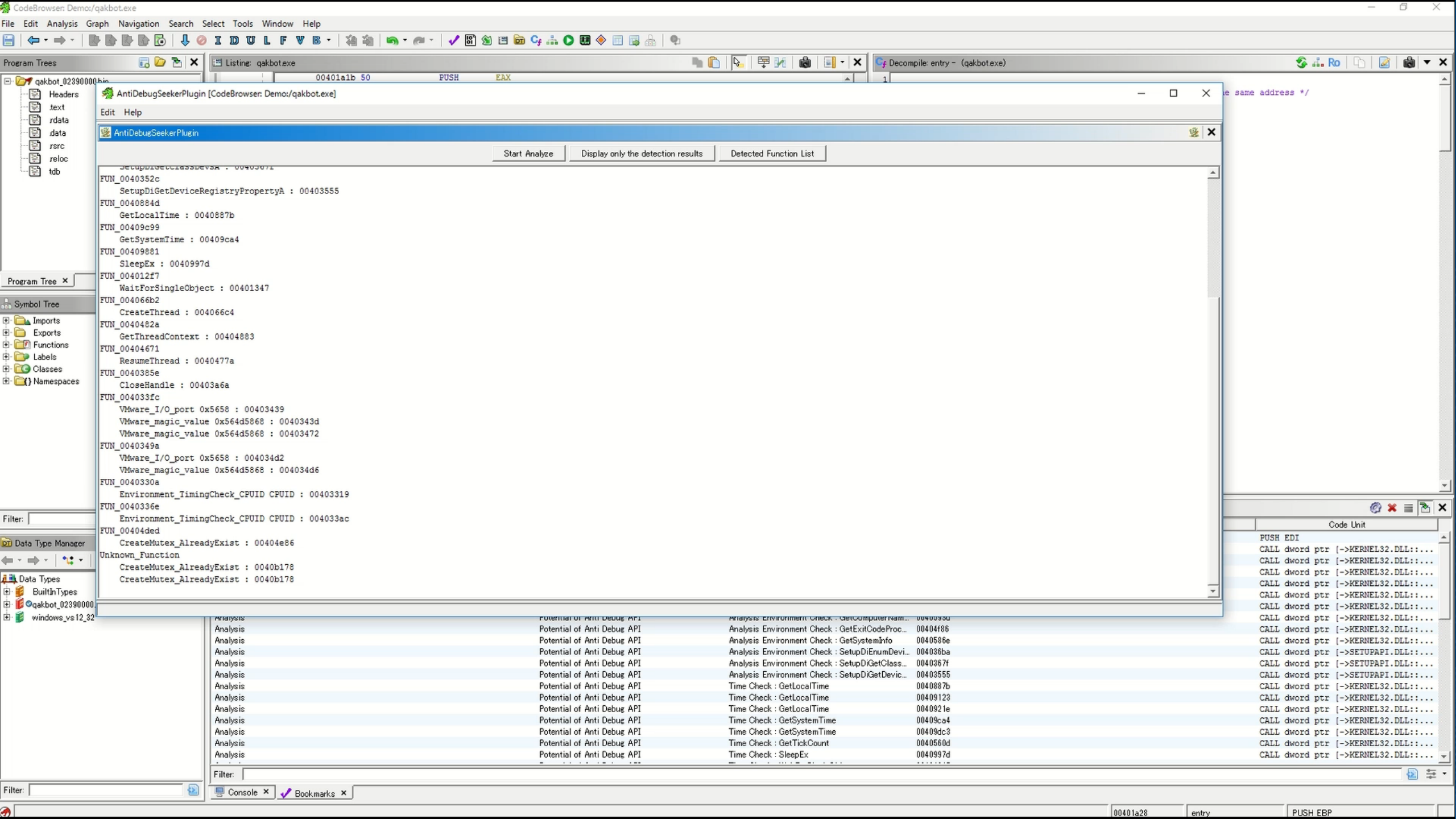
Tree View Table View

Running Tools

Workspace

Fri Jun 07 16:37:25 JST 2024 Recovery snapshot created: C:\Users\kaihatu\Demo\rep\data\004\00000000.db\snapshotA.grf





# The Analysis result of Ghidra-AntiDebugSeeker

```
if (bVar1) {  
    uVar4 = FUN_004033fc(pCVar3,extraout_EDX);  
    FUN_0040349a(extraout_ECX_00,(int)((ulonglong)uVar4 >> 0x20));  
    FUN_004035b6();  
    FUN_0040385e();  
    FUN_00403bdf();  
    FUN_00403d22();  
    FUN_0040336e();  
}
```

Anti Debug Codes

Only Return

Anti	LAB_00403c14	XREF[1]: 00403c10(j)
FUN	00403c14 ff 15 80 b0 40 00	CALL dword ptr [->KERNEL32.DLL::GetCurrentProcessId] = 0000f798
FUN	00403c1a 50	PUSH EAX
FUN	00403c1b 6a 08	PUSH 0x8
FUN	00403c1d ff 15 88 07 41 00	CALL dword ptr [DAT_00410788]
FUN		call dword ptr ds:[<&GetCurrentProcessId>] push eax push 8 call dword ptr ds:[<&CreateToolhelp32Snapshot>]

FUN_403bdf	No Detected
FUN_403d22	No Detected
FUN_40336e	Detected ( Environment_TimingCheck )



# Introduction to Files related to the Ghidra version

- Ghidra Script

AntiDebugSeeker.java

- Ghidra Extension

Ghidra\_11.0.1\_PUBLIC\_AntiDebugSeeker.zip

- Configuration Files

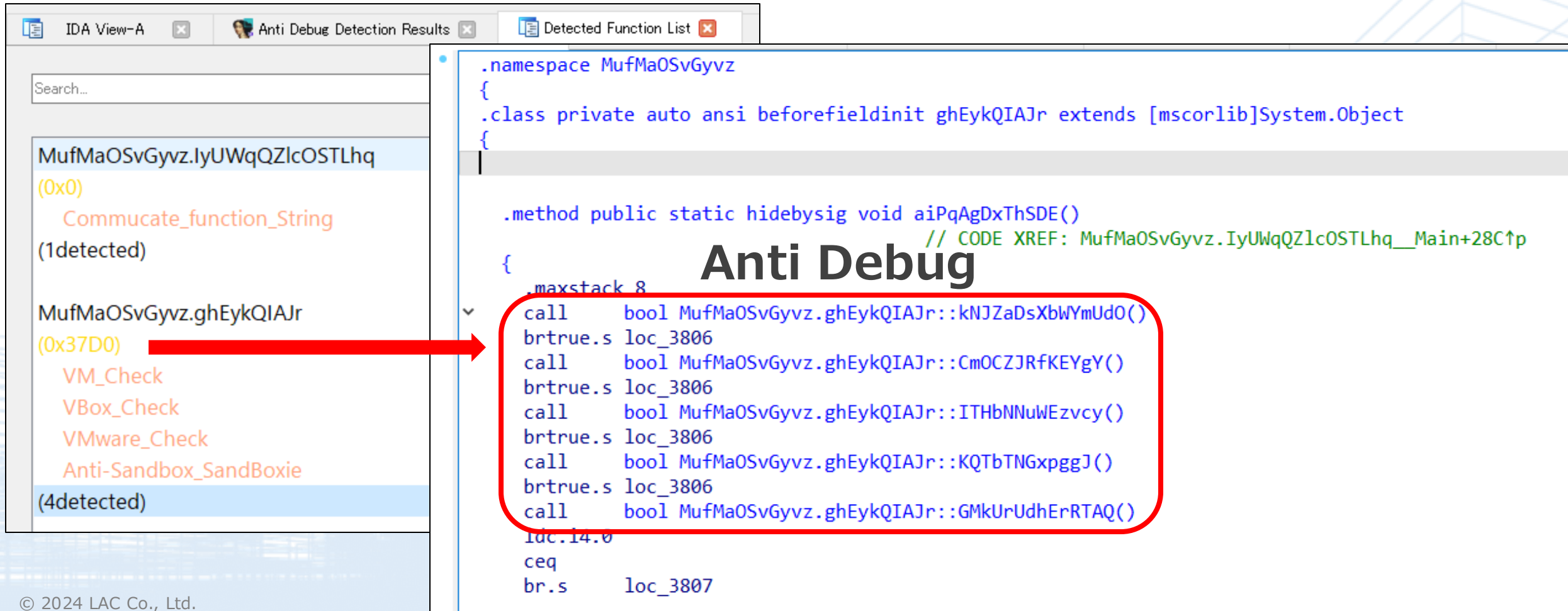
anti\_debug\_Ghidra.config

anti\_debug\_techniques\_descriptions\_Ghidra.json

# AntiDebugSeeker VS .Net Base Malware

## Malware : Thanos Ransomware

❑ MD5 : e01e11dca5e8b08fc8231b1cb6e2048c



The screenshot displays the IDA Pro interface with three windows: 'IDA View-A', 'Anti Debug Detection Results', and 'Detected Function List'. The 'Anti Debug Detection Results' window shows a search for 'MufMaOSvGyvz.ghEykJIAJr' (0x37D0) with four detected functions: VM\_Check, VBox\_Check, VMware\_Check, and Anti-Sandbox\_SandBoxie. The 'Detected Function List' window shows the assembly code for the 'aiPqAgDxThSDE()' method, which is circled in red and labeled 'Anti Debug'. The code includes several calls to other methods in the 'MufMaOSvGyvz.ghEykJIAJr' namespace, such as 'kNJZaDsXbwYmUd0()', 'CmOCZJRfKEYgY()', 'ITHbNNUwEzvcy()', 'KQTbTNGxpggJ()', and 'GMkUrUdhErRTAQ()'. A red arrow points from the 'Anti-Sandbox\_SandBoxie' function in the detection results to the assembly code.

```
.namespace MufMaOSvGyvz
{
.class private auto ansi beforefieldinit ghEykJIAJr extends [mscorlib]System.Object
{
    .method public static hidebysig void aiPqAgDxThSDE()
        // CODE XREF: MufMaOSvGyvz.IyUWqQZlcOSTLhq__Main+28C↑p
    {
        .maxstack 8
        call    bool MufMaOSvGyvz.ghEykJIAJr::kNJZaDsXbwYmUd0()
        brtrue.s loc_3806
        call    bool MufMaOSvGyvz.ghEykJIAJr::CmOCZJRfKEYgY()
        brtrue.s loc_3806
        call    bool MufMaOSvGyvz.ghEykJIAJr::ITHbNNUwEzvcy()
        brtrue.s loc_3806
        call    bool MufMaOSvGyvz.ghEykJIAJr::KQTbTNGxpggJ()
        brtrue.s loc_3806
        call    bool MufMaOSvGyvz.ghEykJIAJr::GMkUrUdhErRTAQ()
        ldc.i4.0
        ceq
        br.s    loc_3807
    }
}
```

```
.method private static hideby sig bool ITHbNNUWEzvcy()
{
    .maxstack 2
    .locals init (native int V0,
                 bool V1)
nop
    .try {
ldstr    aSbiedl1D1l // Anti-Sandbox_SandBoxie - It is checking whether the analysis is being performed in a SandBoxie sandbox.
call     native int MufMaOSvGyvz.ghEyKQIAJr::GetModuleHandle(string string_0)
stloc.0
ldloca.s 0
call     instance int32 [mscorlib]System.IntPtr::ToInt32()
ldc.i4.0
ceq
brtrue.s loc_39AC
```

```
.method public static hideby sig bool LEYLEJpRfEgTMCc()
{
    .maxstack 1
    .locals init (class [System]System.Net.WebRequest V0,
                 bool V1)
ldstr    aHttpsWwwGoogle // Commucate_function_String - Indicating potential communication features, possibly for connecting to a C2 server or detecting analysis environments.
call     class [System]System.Net.WebRequest [System]System.Net.WebRequest::Create(string)
```

```
ldloc.3
ldstr aModel // "Model"
callvirt instance object [System.Management]System.Management.ManagementBaseObject::get_Item(string)
callvirt instance string [mscorlib]System.Object::ToString()
callvirt instance string [mscorlib]System.String::ToUpperInvariant()
ldstr aVirtual // VM_Check - It is possible that the analysis environment is detecting whether it is running on
callvirt instance bool [mscorlib]System.String::Contains(string)
brtrue.s loc_3917
```

```
loc_38EA:
ldloc.s 4
ldstr aVmware // VMware_Check - It is possible that the analysis environment is detecting whether it is running on VMware.
callvirt instance bool [mscorlib]System.String::Contains(string)
brtrue.s loc_3917
```

```
loc_3917:
ldc.i4.0
```

```
ldloc.3
ldstr aModel // "Model"
callvirt instance object [System.Management]System.Management.ManagementBaseObject::get_Item(string)
callvirt instance string [mscorlib]System.Object::ToString()
ldstr aVirtualbox // VBox_Check - It is possible that the analysis environment is detecting whether it is running on VirtualBox.
call bool [mscorlib]System.String::op_Equality(string, string)
ldc.i4.0
ceq
br.s loc_3918
```

- ① **Improve analysis efficiency.**
- ② **Custom rule files.**
- ③ **Further enhance debugging efficiency.**

Thank you!

# Github URL (IDA)

[https://github.com/LAC-Japan/IDA\\_Plugin\\_AntiDebugSeeker](https://github.com/LAC-Japan/IDA_Plugin_AntiDebugSeeker)





# Github URL (GHIDRA)

[https://github.com/LAC-Japan/Ghidra\\_AntiDebugSeeker](https://github.com/LAC-Japan/Ghidra_AntiDebugSeeker)





※本資料は作成時点の情報に基づいており、記載内容は予告なく変更される場合があります。

※本資料に掲載の図は、資料作成用のイメージカットであり、実際とは異なる場合があります。

※本資料は、弊社が提供するサービスや製品などの導入検討のためにご利用いただき、他の目的のためには利用しないようご注意ください。

※ LAC、ラック、JSOC、サイバー救急センターは株式会社ラックの登録商標です。その他記載されている会社名、製品名は一般に各社の商標または登録商標です。