Windows Instrumentation

情報セキュリティ国際会議
CODE BLUE

We are going to investigate a non-malicious sample.

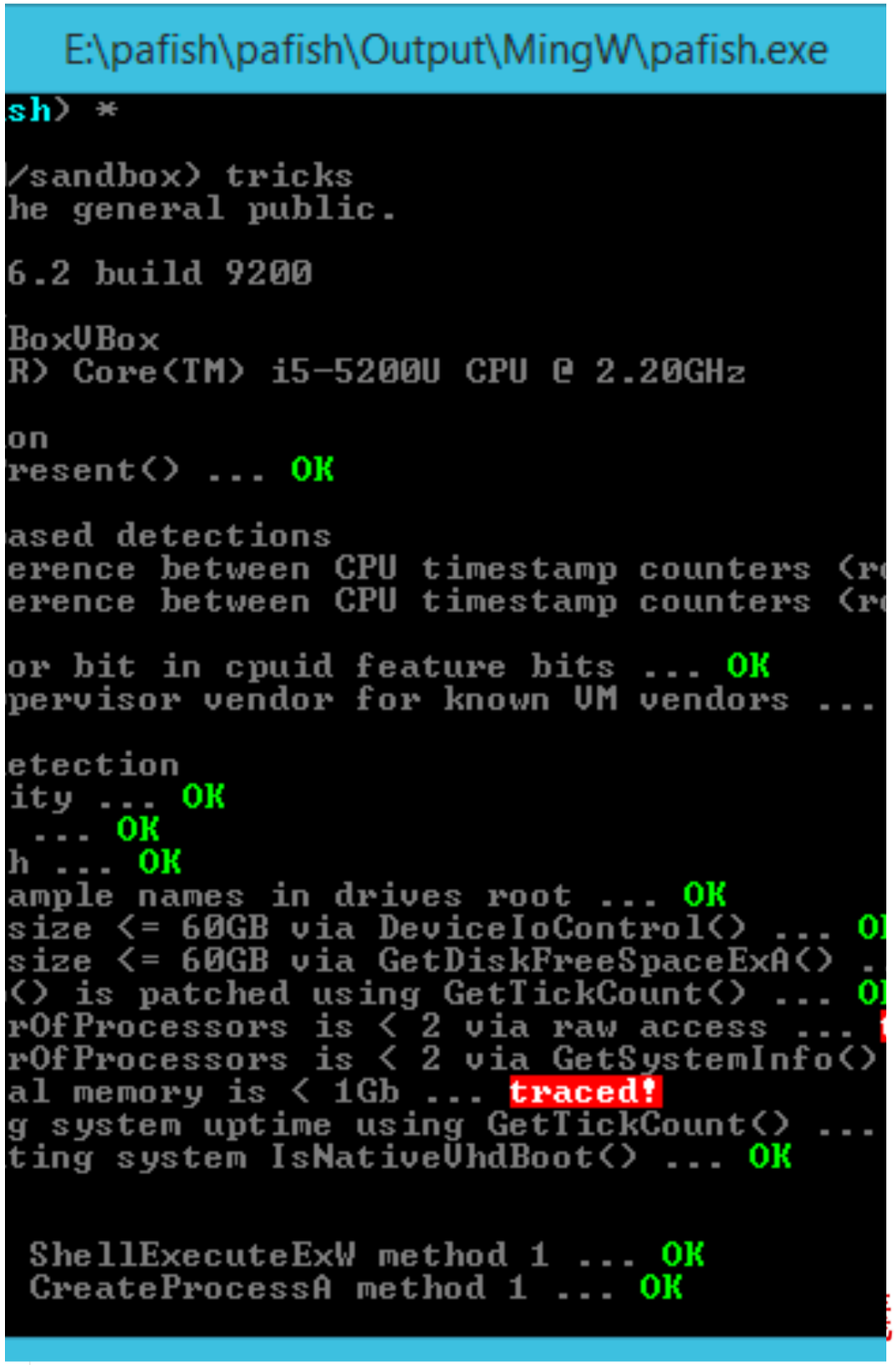
情報セキュリティ国際会議
CODE BLUE

- Malware uses Anti-VM techniques to avoid execution
- These techniques are widespread.
- Tools gather these techniques for researchers AND attackers

- Most of these tools are open-source
- Attackers usually are inspired by these techniques and just copy them into their list of techniques
- Some only do some simple checks, that are enough to avoid execution.

Pafish

Tool that gathers common anti-virtualization techniques



CODE BLUE

What does it cover?

- Virtualization systems: VirtualBox, VMWare, QEMU, KVM, Bochs
- Specific registry checks
- Specific file checks
- Common checks: number of cores, available RAM, disk size…

- For the training purposes, we are not going to have access to the source code.

- Our main goal is to execute the sample in a controlled environment and gather information about its checks

- Patch these checks in runtime to avoid detection

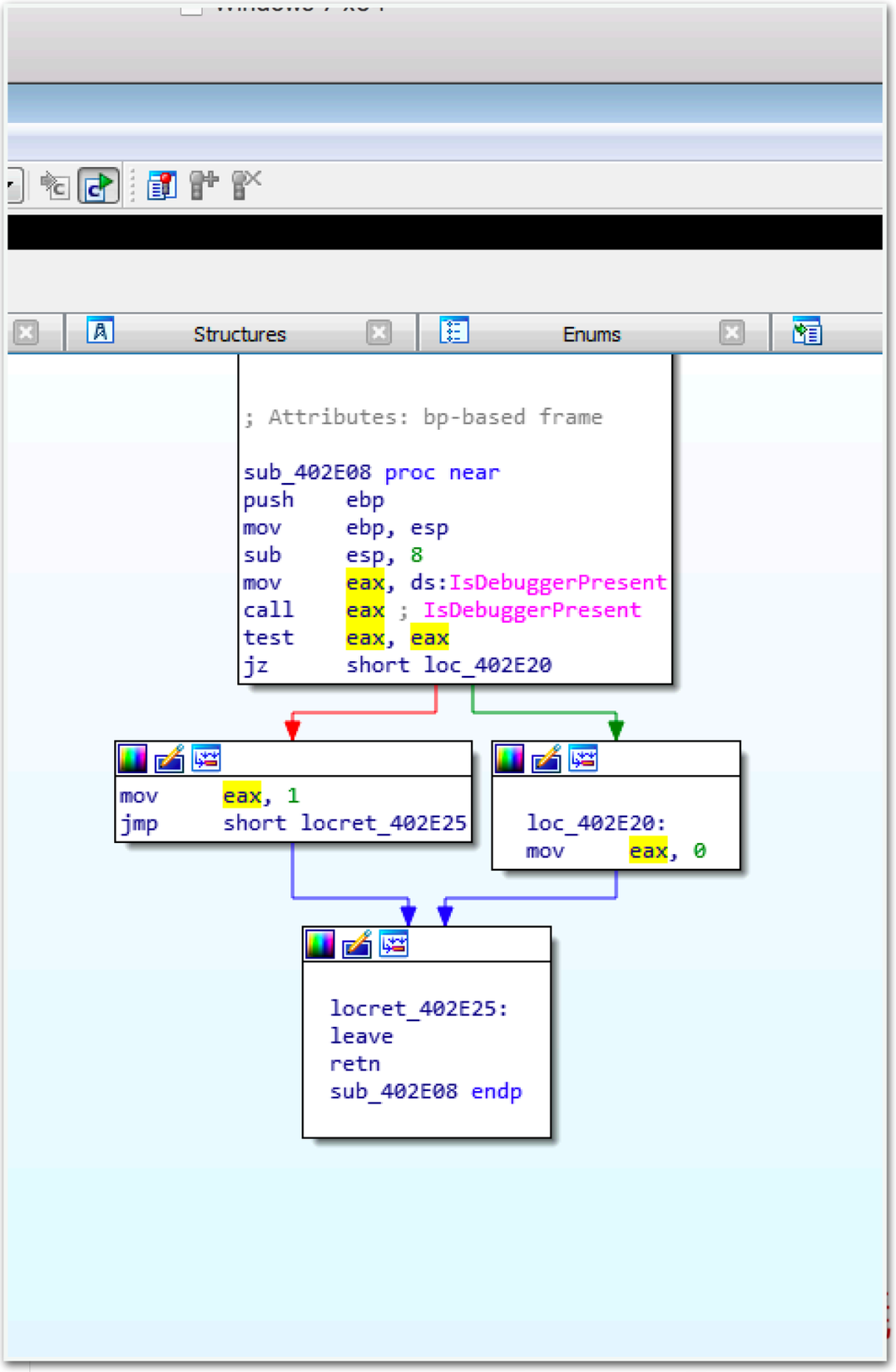
Looks long and hard, doesn’t it?

Actually, it’s easier and shorter - Just use Frida!

IsDebuggerPresent

The application will try to detect if we are trying to detect it.

We can see that in the disassembly, can't we?

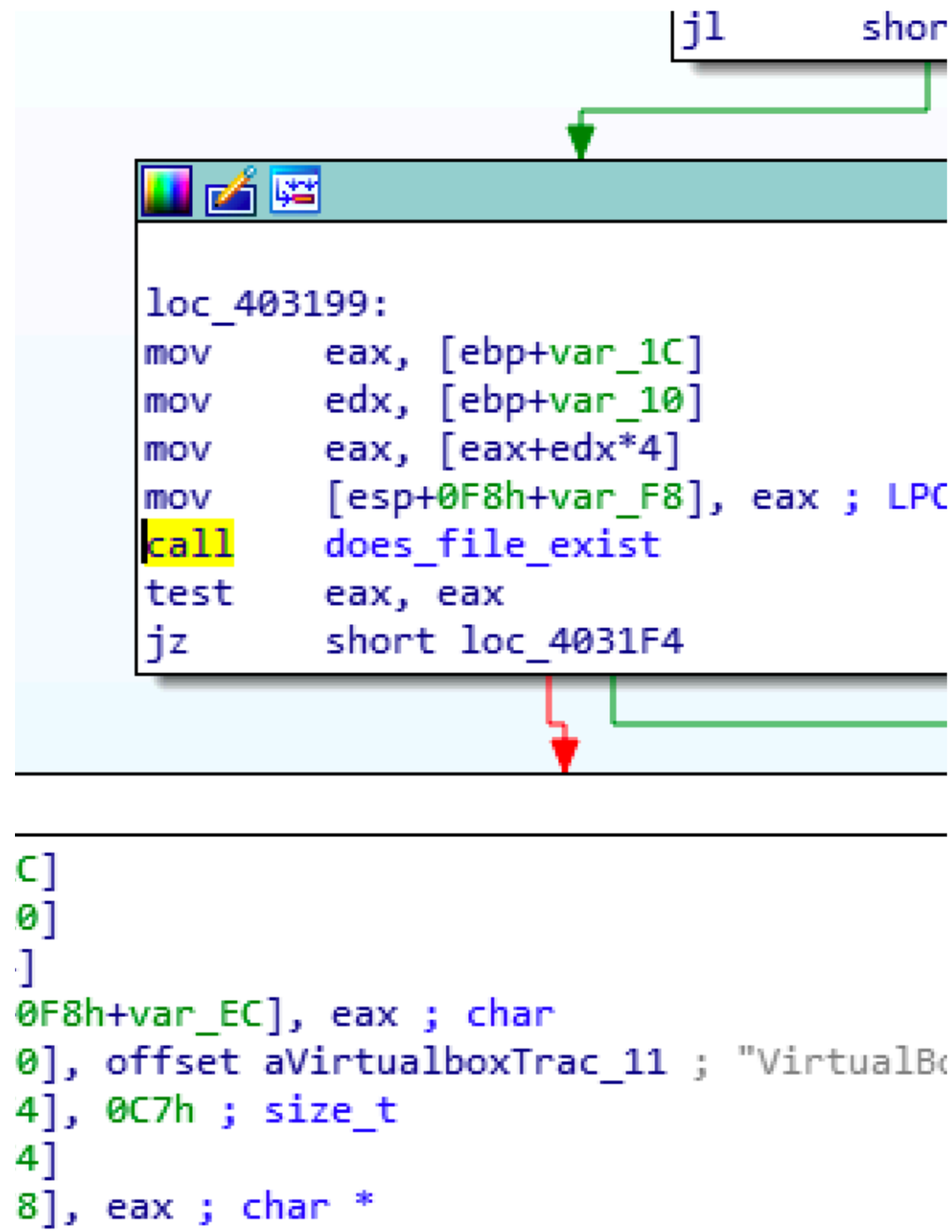


CODE BLUE

Checking for existing files

After reading the disassembly, we find that the API GetFileAttributesA is being called a lot.

Let’s investigate!



CODE BLUE

```
'payload': 'HARDWARE\\ACPI\\RSDT\\VBOX__'}
'payload': 'SYSTEM\\ControlSet001\\Services\\VBoxGuest'}
'payload': 'SYSTEM\\ControlSet001\\Services\\VBoxMouse'}
'payload': 'SYSTEM\\ControlSet001\\Services\\VBoxService'}
'payload': 'SYSTEM\\ControlSet001\\Services\\VBoxSF'}
'payload': 'SYSTEM\\ControlSet001\\Services\\VBoxVideo'}
'payload': 'HARDWARE\\DESCRIPTION\\System'}
'payload': 'C:\\WINDOWS\\system32\\drivers\\VBoxMouse.sys'}
'payload': 'C:\\WINDOWS\\system32\\drivers\\VBoxGuest.sys'}
'payload': 'C:\\WINDOWS\\system32\\drivers\\VBoxSF.sys'}
'payload': 'C:\\WINDOWS\\system32\\drivers\\VBoxVideo.sys'}
'payload': 'C:\\WINDOWS\\system32\\vboxdisp.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxhook.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxmrxnp.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxogl.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxoglarrayspu.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxoglcrutil.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxoglerrorspu.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxoglfeedbackspu.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxoglpackspu.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxoglpassthroughspu.dll'}
'payload': 'C:\\WINDOWS\\system32\\vboxservice.exe'}
'payload': 'C:\\WINDOWS\\system32\\vboxtray.exe'}
'payload': 'C:\\WINDOWS\\system32\\VBoxControl.exe'}
'payload': 'C:\\program files\\oracle\\virtualbox guest additions\\'}
'payload': 'HARDWARE\\DEVICEMAP\\Scsi\\Scsi Port 0\\Scsi Bus 0\\Target Id 0\\Logical Unit Id 0'}
'payload': 'HARDWARE\\DEVICEMAP\\Scsi\\Scsi Port 1\\Scsi Bus 0\\Target Id 0\\Logical Unit Id 0'}
```

We can see the files!

Are there interesting string patterns that we can patch?

Disk Space checks

Pafish is able to recognize if it's being virtualized by using two different methods

Can you figure out at least 1 of them, and patch it?

```
v     [esp+538h+Source], offset aHiSandboxUsern ; "hi_sandbox_
v     [esp+538h+var_530], offset aSandboxTracedB ; "Sandbox tr
v     [esp+538h+var_534], offset sub_403858 ; int
v     [esp+538h+lpVersionInformation], offset aCheckingUserna
ll    sub_402445
v     [esp+538h+Source], offset aHiSandboxPath ; "hi_sandbox_p
v     [esp+538h+var_530], offset aSandboxTracedB_0 ; "Sandbox
v     [esp+538h+var_534], offset sub_40393C ; int
v     [esp+538h+lpVersionInformation], offset aCheckingFilePa
ll    sub_402445
v     [esp+538h+Source], offset aHiSandboxCommo ; "hi_sandbox_
v     [esp+538h+var_530], offset aSandboxTracedB_1 ; "Sandbox
v     [esp+538h+var_534], offset sub_403A22 ; int
v     [esp+538h+lpVersionInformation], offset aCheckingCommon
ll    sub_402445
v     [esp+538h+Source], offset aHiSandboxDrive ; "hi_sandbox_
v     [esp+538h+var_530], offset aSandboxTracedB_2 ; "Sandbox
v     [esp+538h+var_534], offset sub_403B65 ; int
v     [esp+538h+lpVersionInformation], offset aCheckingIfDisk
ll    sub_402445
v     [esp+538h+Source], offset aHiSandboxDrive_0 ; "hi_sandbo
v     [esp+538h+var_530], offset aSandboxTracedB_3 ; "Sandbox
v     [esp+538h+var_534], offset sizeby_getdiskfreespaceexa ;
v     [esp+538h+lpVersionInformation], offset aCheckingIfDisk_
ll    sub_402445
v     [esp+538h+Source], offset aHiSandboxSleep ; "hi_sandbox_
v     [esp+538h+var_530], offset aSandboxTracedB_4 ; "Sandbox
v     [esp+538h+var_534], offset sub_403CAA ; int
v     [esp+538h+lpVersionInformation], offset aCheckingIfSlee
ll    sub_402445
v     [esp+538h+Source], offset aHiSandboxNumbe ; "hi_sandbox_
v     [esp+538h+var_530], offset aSandboxTracedB_5 ; "Sandbox
v     [esp+538h+var_534], offset sub_403CEA ; int
v     [esp+538h+lpVersionInformation], offset aCheckingIfNumb
ll    sub_402445
v     [esp+538h+Source], offset aHiSandboxNumbe_0 ; "hi_sandbo
v     [esp+538h+var_530], offset aSandboxTracedB_6 ; "Sandbox
v     [esp+538h+var_534], offset sub_403D12 ; int
v     [esp+538h+lpVersionInformation], offset aCheckingIfNumb_
ll    sub_402445
v     [esp+538h+Source], offset aHiSandboxPysic ; "hi_sandbox_
v     [esp+538h+var_530], offset aSandboxTracedB_7 ; "Sandbox
v     [esp+538h+var_534], offset sub_403D36 ; int
v     [esp+538h+lpVersionInformation], offset aCheckingIfPysi
```

CODE BLUE

DeviceIOControl

- Pafish states that it’s checking disk size by performing a query to DeviceIoControl
- Before doing this, it generates a handler to \\\\.\\PhysicalDrive0
- A call to CreateFileW() is issued
- We can read and redirect this value. Malware might think that there aren't enough privileges or that something strange is happening

GetFreeDiskSpaceExA

- A second back-up method is called by GetFreeDiskSpaceExA
- IMPORTANT: ULARGE_INTEGER is used, not a regular integer is returned
- We can read and modify it: Memory(readU64,writeU64)

Device RAM

Pafish is also detecting us by getting the available RAM

What shall we do?

```
[*] Checking if pysical memory is < 1Gb ... OK
[*] Checking operating system uptime using GetTickCount() ... OK
[*] Checking if operating system IsNativeVhdBoot() ... OK

[-] Hooks detection
[*] Checking function ShellExecuteExW method 1 ... traced!
[*] Checking function CreateProcessA method 1 ... traced!

[-] Sandboxie detection
[*] Using GetModuleHandle(sbiedll.dll) ... OK
```

LPMEMORYSTATUSEX

- When **GlobalMemoryStatusEx**() is called, a struct is being
- MEMORYSTATUSEX is a struct that contains the following information:

```
typedef struct _MEMORYSTATUSEX {
        DWORD dwLength;
        DWORD dwMemoryLoad;
        DWORDLONG ullTotalPhys;
        DWORDLONG ullAvailPhys;
        DWORDLONG ullTotalPageFile;
        DWORDLONG ullAvailPageFile;
        DWORDLONG ullTotalVirtual;
        DWORDLONG ullAvailVirtual;
        DWORDLONG ullAvailExtendedVirtual;
} MEMORYSTATUSEX, *LPMEMORYSTATUSEX;
```