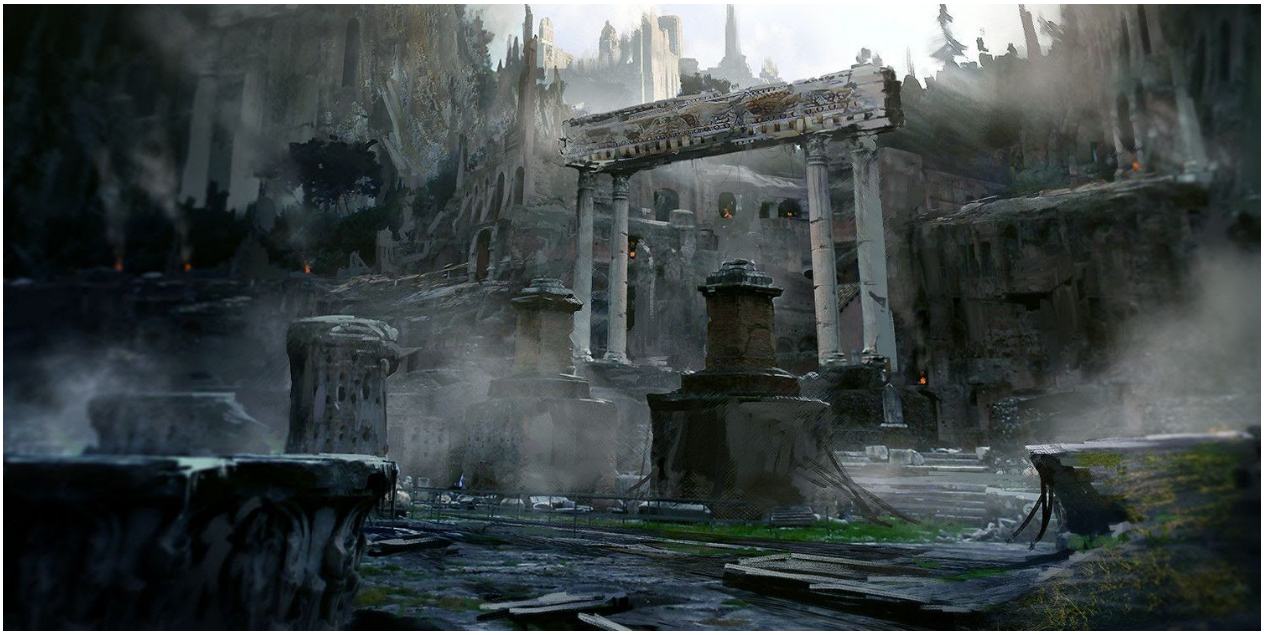
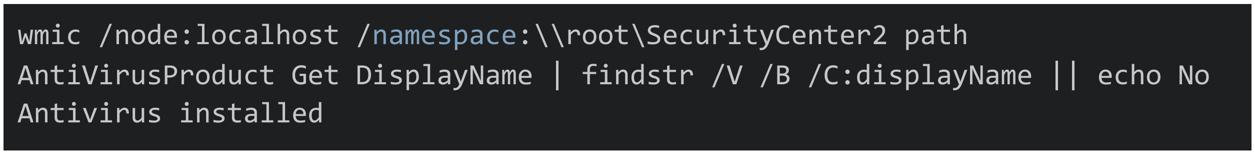
ANTI-VIRUS ARTIFACTS

// By Devisha Rochlani



As 2020 comes to end I have seen many anti-virus evasion methods come and go. Most notably there has been a resurgence of classic anti-hooking techniques (note the release date) which have proven to be effective against many AV and EDR systems. While this is effective a question still remains to be unanswered: if we are being hooked, who is hooking us? The most common method to determine if an anti-virus product or EDR system is in place is using the WMIC and performing a basic query against the Windows Security Center namespace.



```
wmic /node:localhost /namespace:\\root\SecurityCenter2 path
AntiVirusProduct Get DisplayName | findstr /V /B /C:displayName || echo No
Antivirus installed
```

courtesy of Sam Denty from StackOverflow

This method will work in most scenarios. The problem presented here is that this will only return a string if the anti-virus product, or the EDR system, has chosen to register itself in the Windows Security Center namespace. If the product has not registered itself this query will fail. Knowing we are dependent on a security product to register itself I have decided to go down a different path. In this paper I will document antiviral remnants: artifacts present on the machine which can indicate whether or not a security product is in place thus removing our dependency on the Windows Security Center namespace.

In this paper I have reviewed the following anti-viruses:

- Avira: unable to detect hooks, located drivers / minifilters
- F-Secure: unable to detect hooks, located drivers / minifilters

- Norton: 3 hooked DLLs, located drivers / minifilters
- TrendMicro: 3 hooked DLLs, located drivers / minifilters
- WebRoot: 5 hooked DLLs, located drivers / minifilters
- BitDefender: 6 hooked DLLs, located drivers / minifilters
- MalwareBytes: 8 hooked DLLs, located drivers / minifilters

These products were chosen arbitrarily. The number of products chosen was also random. The purpose of this paper is answer the following questions:

1. What functions are being hooked?
2. What (if any) artifacts are present on the system?

Let’s begin.

Avira

Drivers present:

Name	Description	Path
avkmgr.sys	Avira Manager Driver	C:\Windows\System32\Drivers\
avipbb.sys	Avira Driver for Security Enhancement	C:\Windows\System32\Drivers\
avusbflt.sys	Avira USB Filter Driver	C:\Windows\System32\Drivers\
avdevprot.sys	Avira USB Feature Driver	C:\Windows\System32\Drivers\
avnetflt.sys	Avira WFP Network Driver	C:\Windows\System32\Drivers\
avgntflt.sys	Avira Minifilter Driver	C:\Windows\System32\Drivers\

FSecure

Drivers present:

Name	Description	Path
nif2s64.sys	F-Secure NIF2 Core Driver	ProgramFilesx86\F-Secure\Antivirus\Ultralight\..\
fshs.sys	DG 64-bit kernel module	ProgramFilesx86\F-Secure\Antivirus\Ultralight\..\
fsulgk.sys	F-Secure Gatekeeper 64 bit	ProgramFilesx86\F-Secure\Antivirus\Ultralight\..\

Norton

Drivers Present:

Name	Description	Path
BHDrvx64.sys	Bash Driver	ProgramFile\NortonSecurity \NortonData\..\
IDSVia64.sys	IDS Core Driver	ProgramFile\NortonSecurity \NortonData\..\
SymEvnt.sys	Symantec Eventing Platform	ProgramFile\NortonSecurity \NortonData\..\

DLL’s present:

Name	Description	Path
IPSEng32.dll	IPS Script Engine DLL	ProgramFile\NortonSecurity \NortonData\..\

Functions Hooked

KERNELBASE.DLL

VirtualAllocEx	CreateFileMappingW	CreateFileMappingNumaW
CreateFileW	MapViewOfFile	VirtualProtect
HeapCreate	VirtualAlloc	MapViewOfFileEx
CreateRemoteThreadEx	WriteProcessMemory	VirtualProtectEx

NTDLL.DLL

RtlAddVectoredExceptionHandler	RtlRemoveVectoredExceptionHandler	LdrLoadDll
RtlCreateHeap	NtSetInformationProcess	NtMapViewOfSection
NtWriteVirtualMemory	NtCreateSection	NtProtectVirtualMemory
NtCreateFile	NtCreateProcess	NtCreateThreadEx
NtCreateUserProcess	KiUserExceptionDispatcher	N/A

KERNEL32.DLL

CreateFileMappingA	SetProcessDEPPolicy	VirtualAlloc
MapViewOfFile	CreateFileMappingW	VirtualProtect
HeapCreate	MapViewOfFileEx	CreateRemoteThread
VirtualAllocEx	VirtualProtectEx	WriteProcessMemory
WinExec	N/A	N/A

Trend Micro

Drivers Present:

Name	Description	Path
TMBEC64.sys	Trend Micro early boot Driver	C:\Windows\System32\Drivers\
tmnciesc.sys	Trend Micro NICE Scanner	C:\Windows\System32\Drivers\
tmeevw.sys	Trend Micro Eagle Eye Driver	C:\Windows\System32\Drivers\
tmeyes.sys	TrendMicro Eyes driver Module	C:\Windows\System32\Drivers\
TMUMH.sys	Trend Micro UMH Driver x64	C:\Windows\System32\Drivers\
tmusa.sys	Trend Micro Osprey Scanner Driver	C:\Windows\System32\Drivers\

DLL's present:

Name	Description	Path
tmmon64.dll	Trend Micro UMH Monitor Engine	System32\tmumh\20019\..\
TmUmEvt64.dll	Trend Micro User-Mode Hook	System32\tmumh\20019\..\
TmAMSIProvider64.dll	Trend Micro AMSI Provider Module	System32\TmAMSI\TmAMSIProvider64.dll

Functions Hooked

KERNELBASE.DLL

CreateFileA	CreateFileW	LoadLibraryExW
CreateFileMappingW	LoadLibraryExA	CreateRemoteThreadEx
VirtualAlloc	MapViewOfFile	VirtualProtect
HeapCreate	WriteProcessMemory	VirtualProtectEx
LoadLibraryA	LoadLibraryW	N/A

KERNEL32.DLL

CreateFileMappingA	N/A	N/A

NTDLL.DLL

RtlCreateHeap	LdrUnloadDll	LdrUnloadDll
NtMapViewOfSection	NtUnmapViewOfSection	NtContinue
NtCreateSection	NtProtectVirtualMemory	NtCreateFile
NtSetContextThread	N/A	N/A

WebRoot

Driver's Present:

Name	Description	Path
WRkm.sys	Webroot SecureAnywhere	C:\Windows\System32\Drivers\

DLL's present:

Name	Description	Path
WRusr.dll	Webroot SecureAnywhere	Windows\SysWOW64\WRusr.dll

Functions Hooked

ADVAPI32.DLL

OpenSCManagerW	OpenServiceW	OpenSCManagerA
StartServiceW	ControlService	CreateServiceA
CreateServiceW	DeleteService	OpenServiceA
StartServiceA	WmiExecuteMethodW	N/A

USER32.DLL

PostThreadMessageA	PostMessageA	SendMessageA
SendMessageTimeoutA	SetWindowTextA	CreateWindowExA
SetWindowsHookExA	DrawTextExW	CreateWindowExW
PostMessageW	SendMessageW	SetWindowTextW
PostThreadMessageW	SendMessageTimeoutW	SetWindowsHookExW
SetWinEventHook	SendMessageCallbackW	SendNotifyMessageW
ExitWindowsEx	MessageBoxTimeoutW	SendMessageCallbackA

KERNELBASE.DLL

OutputDebugStringA	CreateProcessInternalW	N/A

NTDLL.DLL

NtWaitForSingleObject	NtDeviceIoControlFile	NtRequestWaitReplyPort
NtOpenProcess	NtMapViewOfSection	NtTerminateProcess
NtDelayExecution	NtWriteVirtualMemory	NtOpenEvent
NtAdjustPrivilegesToken	NtQueueApcThread	NtCreateEvent
NtCreateSection	NtCreateThread	NtProtectVirtualMemory
NtTerminateThread	NtWaitForMultipleObjects	NtSetValueKey
NtAlpcConnectPort	NtAlpcCreatePort	NtAlpcCreatePortSection
NtAlpcCreateSectionView	NtAlpcSendWaitReceivePort	NtAssignProcessToJobObject
NtConnectPort	NtCreateMutant	NtCreatePort
NtCreateSemaphore	NtCreateThreadEx	NtDeleteKey
NtDeleteValueKey	NtMakeTemporaryObject	NtOpenMutant
NtOpenSemaphore	NtOpenThread	NtQueueApcThreadEx
NtRequestPort	NtSecureConnectPort	NtSetContextThread
NtShutdownSystem	NtSystemDebugControl	CsrClientCallServer

URLMON.DLL

URLDownloadToFileW	URLDownloadToFileA	N/A

WININET.DLL

InternetOpenA	InternetCloseHandle	InternetOpenUrlA

GDI32.DLL

BitBlt	TextOutW	N/A

KERNEL32.DLL

GetTickCount	N/A	N/A

RPCRT4.DLL

RpcSend	RpcSendReceive	NdrSendReceive

BitDefender

Drivers present:

Name	Description	Path
atc.sys	BitDefender Active Threat Controller	C:\Windows\System32\Drivers\

DLLs present:

Name	Description	Path
bdhkm64.dll	BitDefender Hooking DLL	Program Files\BitDefender Antivirus Free\bdkdm\...\
atcuf64.dll	BitDefender Active Threat Controller	Program Files\BitDefender Antivirus Free\atcuf\...\

Functions Hooked

KERNELBASE.DLL

DefineDosDeviceW	CreateProcessW	CreateProcessA
CreateProcessInternalA	CreateProcessInternalW	PeekConsoleInputW
CloseHandle	DeleteFileW	OpenThread
CreateRemoteThreadEx	GetProcAddress	MoveFileWithProgressW
MoveFileExW	GetModuleBaseNameW	GetModuleInformation
GetModuleFileNameExW	EnumProcessModules	SetEnvironmentVariableW
EnumDeviceDrivers	SetEnvironmentVariableA	QueueUserAPC
GetLogicalProcessorInformationEx	LoadLibraryA	LoadLibraryW
GetLogicalProcessorInformation	GetApplicationRecoveryCallback	EnumProcessModulesEx
PeekConsoleInputA	ReadConsoleInputA	ReadConsoleInputW
GenerateConsoleCtrlEvent	ReadConsoleA	ReadConsoleW
CreateRemoteThread	N/A	N/A

USER32.DLL

SetWindowsHookExW	CallNextHookEx	FindWindowExA
SendMessageA	PeekMessageA	PeekMessageW
GetDesktopWindow	SendMessageW	SetWindowLongW
GetKeyState	PostMessageW	EnumDesktopWindows
EnumWindows	GetMessageW	SystemParametersInfoW
FindWindowW	GetAsyncKeyState	SetPropW
FindWindowExW	GetDC	GetMessageA
SystemParametersInfoA	SendNotifyMessageW	SetWinEventHook
PostMessageA	UnhookWindowsHookEx	GetClipboardData
SetWindowLongA	SetClipboardData	SendNotifyMessageA
GetDCEx	GetKeyboardState	GetRawInputData
GetWindowDC	RegisterRawInputDevices	SetWindowsHookExA
FindWindowA	SetPropA	N/A

COMBASE.DLL

CoCreateInstance	CoGetClassObject	N/A

KERNEL32.DLL

Process32NextW	CreateToolhelp32Snapshot	MoveFileExA
MoveFileWithProgressA	DefineDosDeviceA	N/A

GDI32.DLL

CreateDCW	BitBlt	CreateCompatibleDC
CreateBitmap	CreateDCA	CreateCompatibleBitmap

NTDLL.DLL

RtlImageNtHeaderEx	NtSetInformationThread	NtClose
NtOpenProcess	NtMapViewOfSection	NtUnmapViewOfSection
NtTerminateProcess	NtWriteVirtualMemory	NtDuplicateObject
NtReadVirtualMemory	NtAdjustPrivilegesToken	NtQueueApcThread
NtCreateProcessEx	NtCreateThread	NtResumeThread
NtAlpcConnectPort	NtAlpcCreatePort	NtAlpcSendWaitReceivePort
NtCreateProcess	NtCreateThreadEx	NtCreateUserProcess
NtQuerySystemEnvironmentValueEx	NtRaiseHardError	NtSetContextThread
NtSetSystemEnvironmentValueEx	RtlWow64SetThreadContext	RtlReportException

MalwareBytes

Drivers present:

Name	Description	Path
mbae64.sys	MalwareBytes Anti exploit	C:\Windows\System32\Drivers\
farft.sys	MalwareBytes Bytes Antiransomware	C:\Windows\System32\Drivers\
MbamChameleon.sys	MalwareBytes Chameleon	C:\Windows\System32\Drivers\
mbam.sys	MalwareBytes Real Time Protection	C:\Windows\System32\Drivers\
mbamswissarmy.sys	MalwareBytes SwissArmy	C:\Windows\System32\Drivers\
mwac.sys	MalwareBytes Web Protection	C:\Windows\System32\Drivers\

DLL’s present:

Name	Description	Path
mbae.dll	MalwareBytes Anti-exploit	Program Files\MalwareBytes\AntiMalware\mbae.dll

Functions Hooked

MSCVRT.DLL

_wsystem	system	N/A

WSA_32.DLL

WSAStartup	N/A	N/A

SHELL32.DLL

ShellExecuteW	ShellExecuteExW	N/A

NTDLL.DLL

ResolveDelayLoadedAPI	GetDllHandle	CreateProcessInternalW
NtAllocateVirtualMemory	NtProtectVirtualMemory	N/A

KERNELBASE.DLL

VirtualAllocEx	CreateProcessW	CreateProcessInternalW
GetModuleHandleW	CreateFileW	LoadLibraryExW
VirtualProtect	HeapCreate	VirtualAlloc
WriteProcessMemory	CreateFileA	VirtualProtectEx
CreateProcessA	CreateProcessInternalA	N/A

URLMON.DLL

URLDownloadToFileW	URLDownloadToCacheFileA	URLDownloadToCacheFileW
URLDownloadToFileA	URLOpenBlockingStreamA	URLOpenBlockingStreamW
URLOpenStreamA	URLOpenStreamW	N/A

WININET.DLL

InternetReadFile	InternetReadFileExW	HttpOpenRequestW
HttpSendRequestW	HttpSendRequestExW	HttpSendRequestA
HttpSendRequestExA	InternetOpenUrlA	InternetOpenUrlW
HttpOpenRequestA	N/A	N/A

KERNEL32.DLL

SetProcessDEPPolicy	CopyFileA	MoveFileA
MoveFileW	CopyFileW	WinExec

Conclusion:

In this paper you can see a clear trend:

- Some AVs rely on archaic malware methods and technologies.
- Some AVs here do not monitor web traffic. If they do monitor web traffic it must be installed as an entirely separate application for an additional cost to the consumer.
- Some AVs fail to monitor API forwards thus presenting the opportunity to malware authors to invoke functionality from NTDLL rather than the KERNELBASE and/or KERNEL32 sister DLL allowing a complete bypass for the API hook.

Unfortunately, it appears many AVs rely on YARA rules, or something YARA like to perform rudimentary static binary analysis.

As a final note: during testing of this paper I decided to test AVs against a keylogger I had developed. The keylogger, dubbed 'UnderTaker.exe', used RegisterRawInputDevices and GetRawInputData from the USER32.dll to keylog user input. In our test scenarios it evaded virtually every AV except **F-Secure** and **BitDefender**. Also note that BitDefender is the only AV in this list to monitor the RegisterRawInputDevices and GetRawInputData function.

Proof-of-concept IOC:
2a419d2ddf31ee89a8deda913abf1b25d45bb0dc59a93c606756cfa66acb0791