

Sandbox Detection

written by Mert SARICA | 2 December 2019

In my blog posts that I wrote 8-9 years ago (Anti Analiz, Anti Anti-VMWare), I mentioned that malicious individuals who develop malware use various methods to make it difficult for security researchers or systems to analyze their malware on virtual systems.

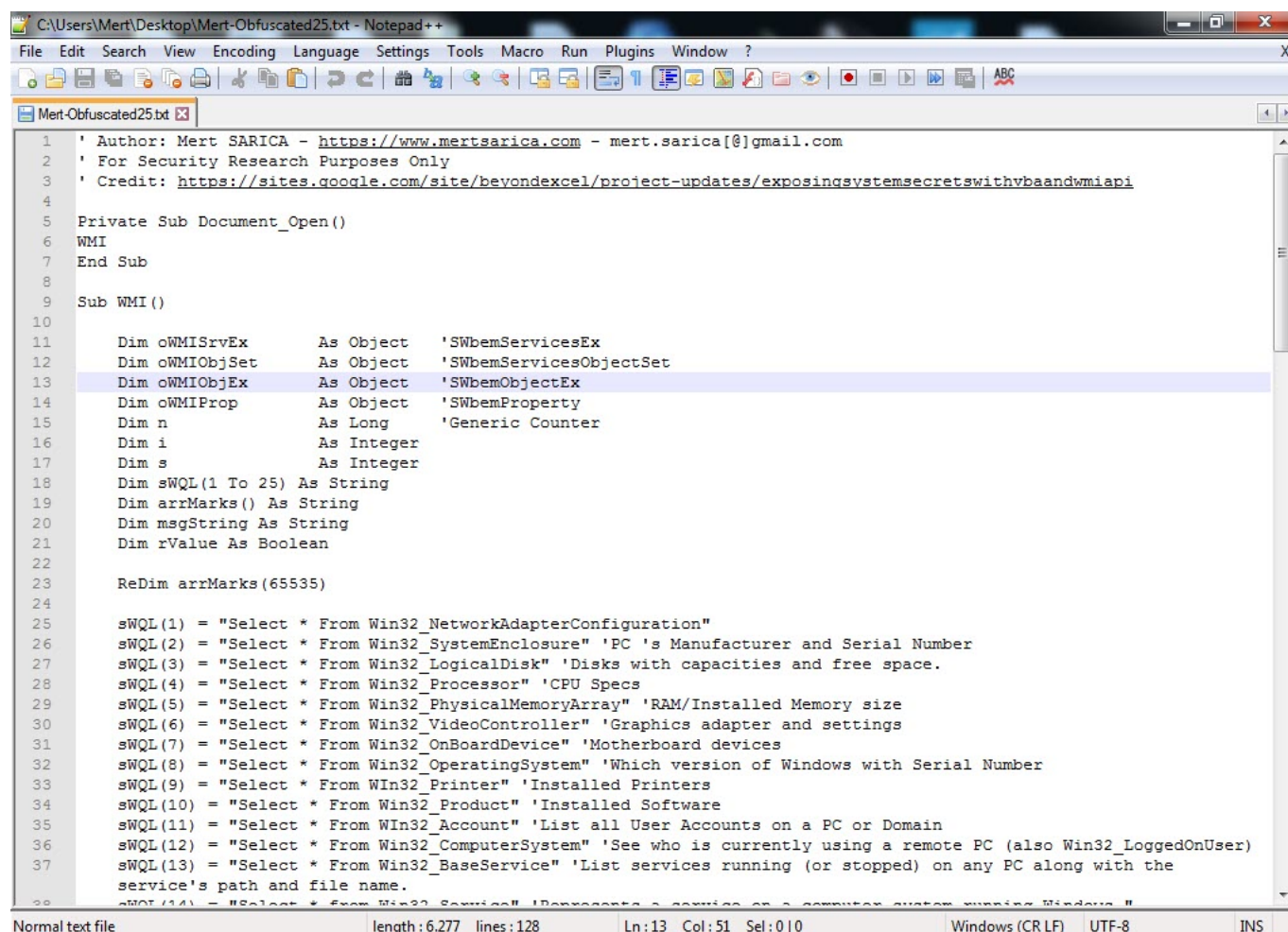
Nowadays, with the widespread use of Virtual Desktop Infrastructure (VDI) technologies in corporate environments, virtual systems are no longer primarily used by servers or malware analysts, security researchers. As a result, malware developers, and also red team members who perform ethical hacking, aim to design and develop tools that can operate on virtual systems but are not detectable by virtual analysis systems. Knowing that it is impossible to develop a tool that does not work on virtual analysis systems with a realistic approach, malware developers are searching for their malware's hash values on VirusTotal at certain intervals to understand if they have been detected and to stop their operations. Similarly, red team members who do not want to be caught, use projects like RedELK to ensure the sustainability of their operations.

I have decided to research and share with you how easy or difficult it is to detect these trusted sandbox systems, such as VirusTotal, Any.Run, Hybrid Analysis, Lastline Analyst, VMRay Analyzer, etc. which are commonly used by end-users and security experts to upload files suspected of being malicious.

To do this, I first needed to gather information (reconnaissance) about the sandbox systems. When a software is uploaded to a sandbox system, it is monitored and recorded by the system when it communicates with a target system (C&C) during dynamic analysis. In short, these systems are allowed to have internet connections on them. So, I decided to prepare a Microsoft Office macro that collects information about the target operating system. The easiest way to do this with the macro is to take advantage of the Windows Management Instrumentation (WMI) which is commonly used for lateral movement in targeted attacks (APT). If you do a small research on Microsoft's website about WMI, you can see that you can collect a lot of information about the target operating system using the Win32 Provider.

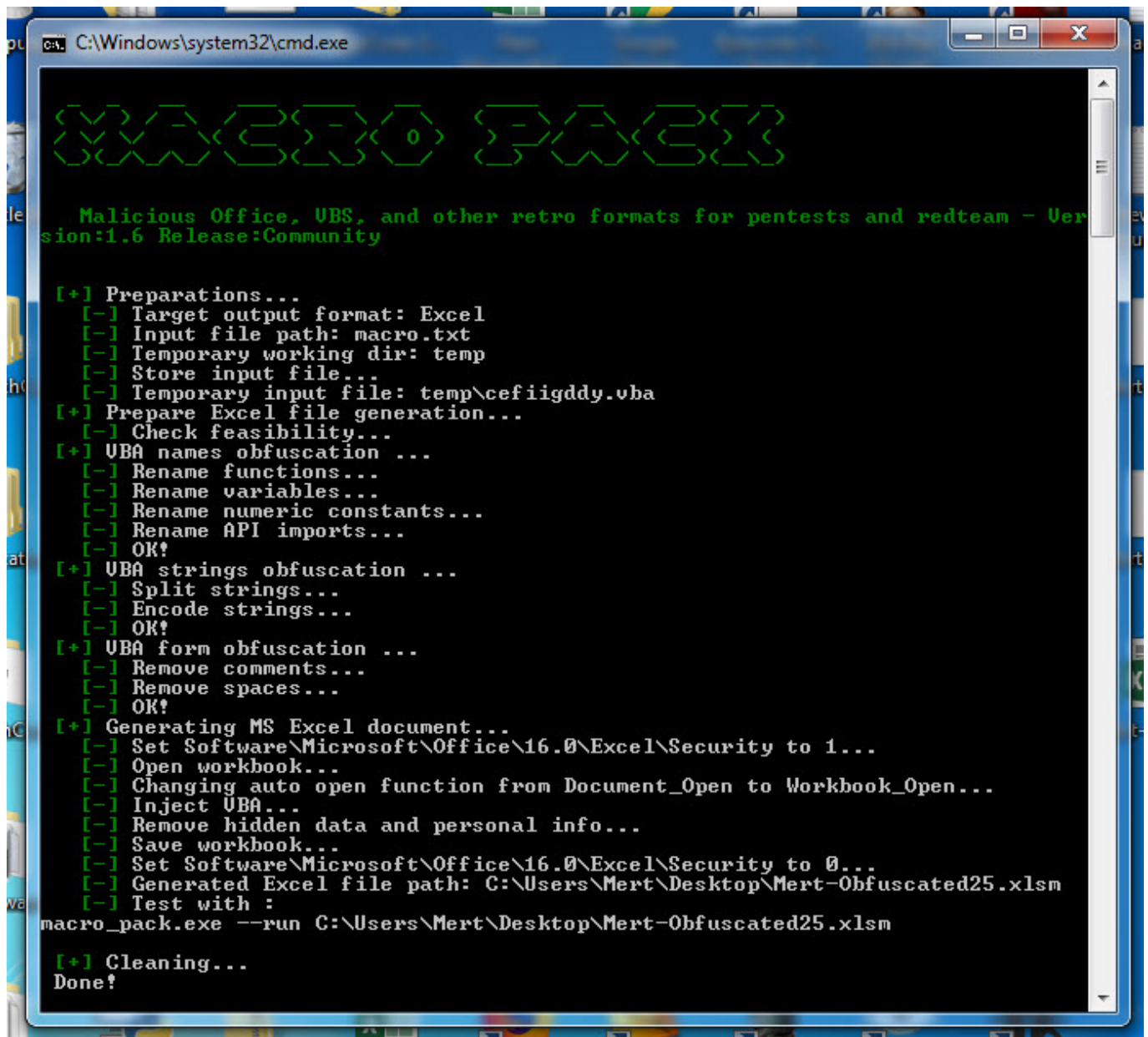
To avoid reinventing the wheel, I did a quick search on Google and came across a simple script that collects information via WMI using VBA. After

adding 25 classes that are specific to the operating system on Microsoft's page to this script file, I made it send the information to <https://www.mertsarica.com/macro.php>. To make sure it cannot be easily detected by antivirus software, I also used the macro_pack tool to hide the macro (obfuscation).

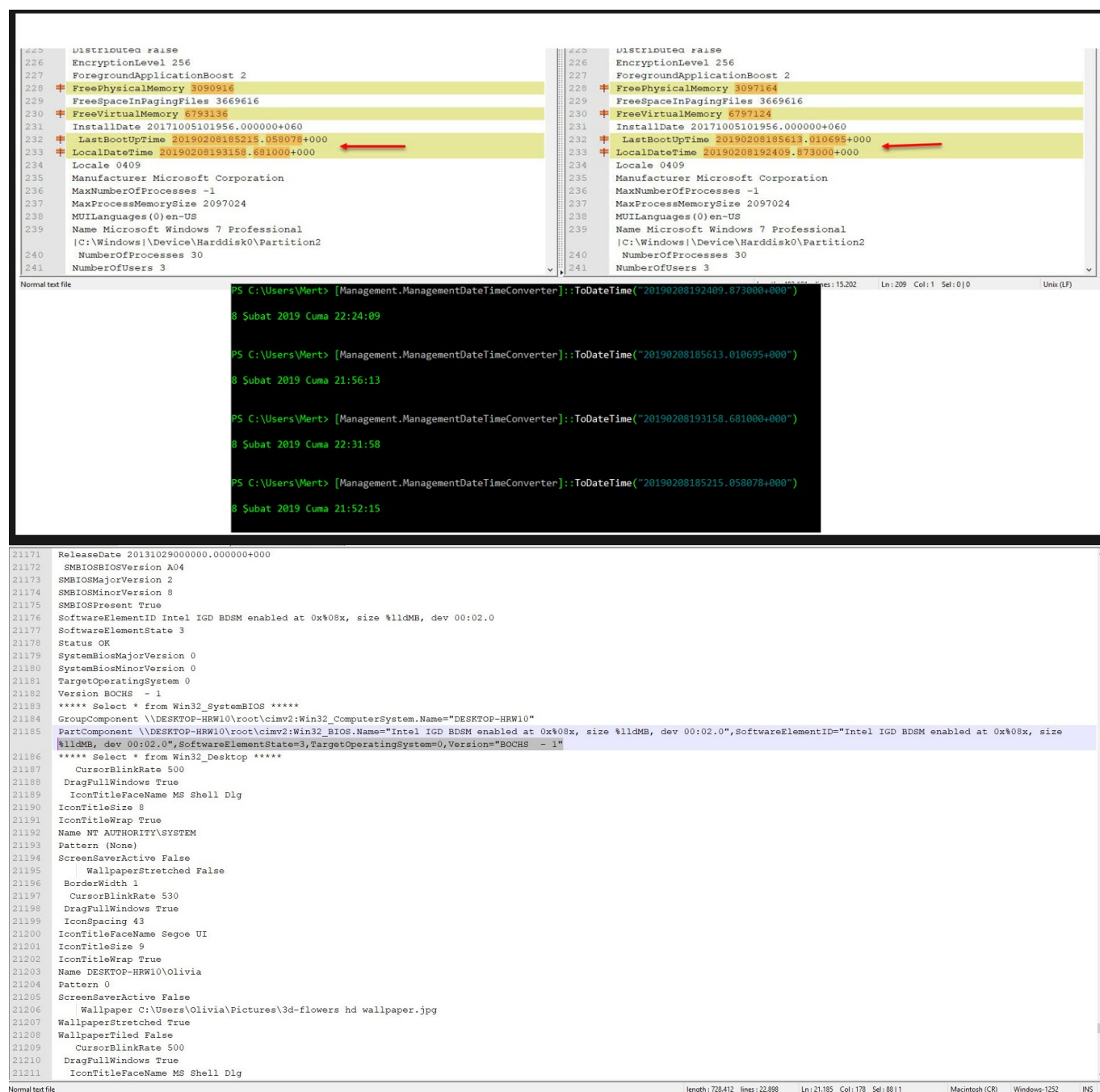


```
1 ' Author: Mert SARICA - https://www.mertsarica.com - mert.sarica[gmail.com]  
2 ' For Security Research Purposes Only  
3 ' Credit: https://sites.google.com/site/bevondexcel/project-updates/exposingsystemsecretswithvbaandwmiapi  
4  
5 Private Sub Document_Open()  
6 WMI  
7 End Sub  
8  
9 Sub WMI()  
10  
11 Dim oWMISrvEx As Object 'SWbemServicesEx  
12 Dim oWMIObjSet As Object 'SWbemServicesObjectSet  
13 Dim oWMIObjEx As Object 'SWbemObjectEx  
14 Dim oWMIProp As Object 'SWbemProperty  
15 Dim n As Long 'Generic Counter  
16 Dim i As Integer  
17 Dim s As Integer  
18 Dim sWQL(1 To 25) As String  
19 Dim arrMarks() As String  
20 Dim msgString As String  
21 Dim rValue As Boolean  
22  
23 ReDim arrMarks(65535)  
24  
25 sWQL(1) = "Select * From Win32_NetworkAdapterConfiguration"  
26 sWQL(2) = "Select * From Win32_SystemEnclosure" 'PC 's Manufacturer and Serial Number  
27 sWQL(3) = "Select * From Win32_LogicalDisk" 'Disks with capacities and free space.  
28 sWQL(4) = "Select * From Win32_Processor" 'CPU Specs  
29 sWQL(5) = "Select * From Win32_PhysicalMemoryArray" 'RAM/Installed Memory size  
30 sWQL(6) = "Select * From Win32_VideoController" 'Graphics adapter and settings  
31 sWQL(7) = "Select * From Win32_OnBoardDevice" 'Motherboard devices  
32 sWQL(8) = "Select * From Win32_OperatingSystem" 'Which version of Windows with Serial Number  
33 sWQL(9) = "Select * From Win32_Printer" 'Installed Printers  
34 sWQL(10) = "Select * From Win32_Product" 'Installed Software  
35 sWQL(11) = "Select * From Win32_Account" 'List all User Accounts on a PC or Domain  
36 sWQL(12) = "Select * From Win32_ComputerSystem" 'See who is currently using a remote PC (also Win32_LoggedOnUser)  
37 sWQL(13) = "Select * From Win32_BaseService" 'List services running (or stopped) on any PC along with the  
38 service's path and file name.  
39 sWQL(14) = "Select * From Win32_Service" 'List all services on a computer system running Windows"
```

Normal text file length: 6,277 lines: 128 Ln: 13 Col: 51 Sel: 0 | 0 Windows (CR LF) UTF-8 INS



operating system from scratch, in its clean state, before analyzing the malware, so there is a maximum time difference of 30 minutes between the date and time of the operating system's reboot (LastBootUpTime) and the date and time of the analysis (LocalDateTime). Based on this information, it is possible to assume that the software was analyzed in the sandbox.



```
226 Distributed false
227 EncryptionLevel 256
228 ForegroundApplicationBoost 2
229 FreePhysicalMemory 3090916
230 FreeSpaceInPagingFiles 3669616
231 FreeVirtualMemory 6793136
232 InstallDate 20171005101956.000000+060
233 LastBootUpTime 20190208185215.058078+000
234 LocalDateTime 20190208193158.681000+000
235 Locale 0409
236 Manufacturer Microsoft Corporation
237 MaxNumberOfProcesses -1
238 MaxProcessMemorySize 2097024
239 MUILanguages(0)en-US
240 Name Microsoft Windows 7 Professional
241 NumberOfProcesses 30
242 NumberOfUsers 3
```

```
PS C:\Users\Mert> [Management.ManagementDateTimeConverter]::ToDateTime("20190208192409.873000+000")
8 Subat 2019 Cuma 22:24:09

PS C:\Users\Mert> [Management.ManagementDateTimeConverter]::ToDateTime("20190208185613.010695+000")
8 Subat 2019 Cuma 21:56:13

PS C:\Users\Mert> [Management.ManagementDateTimeConverter]::ToDateTime("20190208193158.681000+000")
8 Subat 2019 Cuma 22:31:58

PS C:\Users\Mert> [Management.ManagementDateTimeConverter]::ToDateTime("20190208185215.058078+000")
8 Subat 2019 Cuma 21:52:15
```

```
21171 ReleaseDate 20131029000000.000000+000
21172 SMBIOSVersion A04
21173 SMBIOSMajorVersion 2
21174 SMBIOSMinorVersion 8
21175 SMBIOSPresent True
21176 SoftwareElementID Intel IGD BDSM enabled at 0x08x, size 11dMB, dev 00:02.0
21177 SoftwareElementState 3
21178 Status OK
21179 SystemBiosMajorVersion 0
21180 SystemBiosMinorVersion 0
21181 TargetOperatingSystem 0
21182 Version BOCHS - 1
21183 ***** Select * from Win32_SystemBIOS *****
21184 GroupComponent \\DESKTOP-HRW10\root\cimv2:Win32_ComputerSystem.Name="DESKTOP-HRW10"
21185 PartComponent \\DESKTOP-HRW10\root\cimv2:Win32_BIOS.Name="Intel IGD BDSM enabled at 0x08x, size 11dMB, dev 00:02.0,SoftwareElementID="Intel IGD BDSM enabled at 0x08x, size 11dMB, dev 00:02.0",SoftwareElementState=3,TargetOperatingSystem=0,Version="BOCHS - 1"
21186 ***** Select * from Win32_Desktop *****
21187 CursorBlinkRate 500
21188 DragFullWindows True
21189 IconTitleFaceName MS Shell Dlg
21190 IconTitleSize 8
21191 IconTitleWrap True
21192 Name NT AUTHORITY\SYSTEM
21193 Pattern (None)
21194 ScreenSaverActive False
21195 WallpaperStretched False
21196 BorderWidth 1
21197 CursorBlinkRate 530
21198 DragFullWindows True
21199 IconSpacing 43
21200 IconTitleFaceName Segoe UI
21201 IconTitleSize 9
21202 IconTitleWrap True
21203 Name DESKTOP-HRW10\Olivia
21204 Pattern 0
21205 ScreenSaverActive False
21206 Wallpaper C:\Users\Olivia\Pictures\3d-flowers hd wallpaper.jpg
21207 WallpaperStretched True
21208 WallpaperTiled False
21209 CursorBlinkRate 500
21210 DragFullWindows True
21211 IconTitleFaceName MS Shell Dlg
```

As I continued to look at the information I had collected, I came across an output where I noticed a difference of 4 months between LastBootUpTime and LocalDateTime. This raised suspicion since a user system (Windows 7) that hasn't been restarted for 4 months is quite unusual, so I began to investigate this information more closely. As it is known, most security researchers, malware analysts have an isolated, virtual analysis system. To

save time, this analysis system is not restarted each time, but instead, is returned from an instant image (snapshot). An operating system returned from an instant image, LastBootUpTime gradually becomes older, and the time difference between LocalDateTime and it can sometimes be months when a malware is being analyzed. In light of this information, I also checked the WMI section where I suspected that this output had collected information on program groups in the Windows operating system, Win32_LogicalProgramGroup, and this time I saw that the system had tools such as Immunity Debugger, Process Hacker, which are frequently used by security researchers, malware analysts. This gave me the information that my Office file was analyzed by a threat hunter. :)

```
289 Caption Microsoft Windows 7 Professional
290 CodeSet 1252
291 CountryCode 1
292 CreationClassName Win32_OperatingSystem
293 CSCreationClassName Win32_ComputerSystem
294 CSDVersion Service Pack 1
295 CSName MARYHILL-PC
296 CurrentTimeZone 60
297 DataExecutionPrevention_32BitApplications True
298 DataExecutionPrevention_Available True
299 DataExecutionPrevention_Drivers True
300 DataExecutionPrevention_SupportPolicy 2
301 Debug False
302 Description
303 Distributed False
304 EncryptionLevel 256
305 ForegroundApplicationBoost 2
306 FreePhysicalMemory 1252420
307 FreeSpaceInPagingFiles 1791456
308 FreeVirtualMemory 2862980
309 InstallDate 20160419151854.000000+120
310 LastBootUpTime 20181113103206.500000+060
311 LocalDateTime 20190329103419.697000+060
312 Locale 0409
313 Manufacturer Microsoft Corporation
314 MaxNumberOfProcesses -1
315 MaxProcessMemorySize 8589934464
316 MUILanguages(0) en-US
317 Name Microsoft Windows 7 Professional |C:\Windows\Device\Harddisk0\Partition2
318 NumberOfProcesses 50
```

```
PS C:\Users\Mert> [Management.ManagementDateTimeConverter]::ToDateTime("20190329103419.697000+060") #LocalDateTime
29 Mart 2019 Cuma 12:34:19

PS C:\Users\Mert> [Management.ManagementDateTimeConverter]::ToDateTime("20181113103206.500000+060") #LastBootUpTime
13 Kasım 2018 Salı 12:32:06
```

```
15750 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15751 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Games"
15752 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15753 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\RxD Hex Editor"
15754 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15755 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Immunity Inc"
15756 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15757 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Immunity Inc\Immunity Debugger"
15758 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15759 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Java"
15760 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15761 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Maintenance"
15762 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15763 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Microsoft Office 2016 Tools"
15764 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15765 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Process Hacker 2"
15766 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15767 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Process Hacker 2\Help and Support"
15768 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15769 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Python 2.7"
15770 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15771 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\Startup"
15772 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15773 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Public:Start Menu\Programs\WinPcap"
15774 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15775 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Mary Hill:Start Menu"
15776 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15777 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Mary Hill:Start Menu\Programs"
15778 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15779 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Mary Hill:Start Menu\Programs\Accessories"
15780 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15781 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Mary Hill:Start Menu\Programs\Accessories\Accessibility"
15782 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15783 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Mary Hill:Start Menu\Programs\Accessories\System Tools"
15784 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15785 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Mary Hill:Start Menu\Programs\Administrative Tools"
15786 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15787 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Mary Hill:Start Menu\Programs\Maintenance"
15788 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15789 Setting \\MARYHILL-PC\root\cimv2:Win32_LogicalProgramGroup.Name="Mary Hill:Start Menu\Programs\Startup"
15790 Element \\MARYHILL-PC\root\cimv2:Win32_ComputerSystem.Name="MARYHILL-PC"
15791 Setting \\MARYHILL-PC\root\cimv2:Win32_TimeZone.StandardName="W. Europe Standard Time"
```

Ava çıkmış tehdit avcısı :)

Lastly, my attention was also caught by the output of the Select * from Win32_SystemBIOS WMI request. When I looked at the information coming from the sandboxes, I saw that one of them was running on the BOCHS emulator and another one was running on the QEMU emulator. Therefore, I understood that these two systems belong to the sandbox system.

```
21171 ReleaseDate 20131029000000.000000+000
21172 SMBIOSBIOSVersion A04
21173 SMBIOSMajorVersion 2
21174 SMBIOSMinorVersion 8
21175 SMBIOSPresent True
21176 SoftwareElementID Intel IGD BDSM enabled at 0x%08x, size %lldMB, dev 00:02:0
21177 SoftwareElementState 3
21178 Status OK
21179 SystemBiosMajorVersion 0
21180 SystemBiosMinorVersion 0
21181 TargetOperatingSystem 0
21182 Version BOCHS - 1
21183 ***** Select * from Win32_SystemBIOS *****
21184 GroupComponent \\DESKTOP-HRW10\\root\\cimv2:Win32_ComputerSystem.Name="DESKTOP-HRW10"
21185 PartComponent \\DESKTOP-HRW10\\root\\cimv2:Win32_BIOS.Name="Intel IGD BDSM enabled at 0x%08x, size %lldMB, dev 00:02:0",SoftwareElementID="Intel IGD BDSM enabled at 0x%08x, size %lldMB, dev 00:02:0",SoftwareElementState=3,TargetOperatingSystem=0,Version="BOCHS - 1"
21186 ***** Select * from Win32_Desktop *****
21187 CursorBlinkRate 500
21188 DragFullWindows True
21189 IconTitleFaceName MS Shell Dlg
21190 IconTitleSize 8
21191 IconTitleWrap True
21192 Name NT AUTHORITY\\SYSTEM
```

Bochs is a portable IA-32 and x86-64 IBM PC compatible emulator and debugger mostly written in C++ and distributed as free software under the GNU Lesser General Public License. It supports emulation of the processor, memory, disks, display, Ethernet, BIOS and common hardware peripherals of PCs













```
C:\Users\Mert\Desktop\Macro\72.12.209.146_1.txt
13771 StartMode Disabled
13776 SystemName PC-4A095E27CB
13800 SystemName PC-4A095E27CB
13819 StartMode Disabled
13824 SystemName PC-4A095E27CB
13848 SystemName PC-4A095E27CB
13867 StartMode Disabled
13872 SystemName PC-4A095E27CB
13876 BiosCharacteristics(17)79
13877 BIOSVersion(1)Intel IGD BDSM enabled at 0x%08x, size %lldMB, dev 00:02:0
13880 Manufacturer SeaBIOS
13884 SMBIOSBIOSVersion rel-1.11.0-0-g63451fca13-prebuilt.qemu-project.org
13892 Version QEMU - 1
13894 GroupComponent \\PC-4A095E27CB\\root\\cimv2:Win32_ComputerSystem.Name="PC-4A095E27CB"
13895 PartComponent \\PC-4A095E27CB\\root\\cimv2:Win32_BIOS.Name="Intel IGD BDSM enabled at 0x%08x, size %lldMB, dev 00:02:0",SoftwareElementID="Intel IGD BDSM enabled at 0x%08x, size %lldMB, dev 00:02:0",SoftwareElementState=3,TargetOperatingSystem=0,Version="QEMU - 1"
13907 Borderwidth 0
13908 CursorBlinkRate 530
13909 DragFullWindows True
13910 IconTitleFaceName Tahoma
13911 IconTitleSize 8
13912 IconTitleWrap True
13913
13914
13915
13916
13917
13918
13919
13920
13921
13922
13923
13924
13925
13926
13927
13928
13929
13930
13931
13932
13933
13934
13935 IconTitleWrap True
13936 Name PC-4A095E27CB\\STRAZNJICA.GRUBUTT
13937 Pattern (None)
13938 ScreenSaverActive False
13939 ScreenSaverSecure False
13940 ScreenSaverTimeout 60
```

QEMU is a free and open-source emulator that performs hardware virtualization. QEMU is a hosted virtual machine monitor: it emulates the machine's processor through dynamic binary

Ln 7807, Col 23 15,377 lines INS Read-only Edit Plug-in Newer 500.2 KB ANSI

Based on the IP addresses that made requests to the macro.php file during the period of time up until October, I can say that these are most likely from VMRay, Lastline, Any.RUN, VirusTotal sandbox systems and a threat hunter's

system.

IP	Domain	Country	Region	City	ISP	ASN	NS
104.215.89.177		 United States	Texas	San Antonio	Microsoft Corporation	8075	
13.80.140.46		 Netherlands	North Holland	Amsterdam	Microsoft Corporation	8075	
188.99.240.204	dsib-188-099-240-204.188.099.pools.vodafone-ip.de	 Germany	Baden-Württemberg Region	Bodman-Ludwigshafen	Vodafone GmbH	3209	
217.86.42.248	pD9562AF8.dip0.t-ipconnect.de	 Germany	Baden-Württemberg Region	Tett nang Castle	Deutsche Telekom AG	3320	
64.233.172.230	google-proxy-64-233-172-230.google.com	 United States			Google LLC	15169	
66.102.6.213	google-proxy-66-102-6-213.google.com	 United States			Google LLC	15169	
66.249.88.41	google-proxy-66-249-88-41.google.com	 United States	California	Mountain View	Google LLC	15169	
66.249.88.60	google-proxy-66-249-88-60.google.com	 United States	California	Mountain View	Google LLC	15169	
71.59.36.230	c-71-59-36-230.hsd1.ga.comcast.net	 United States	Georgia	Atlanta	Comcast Cable Communications, LLC	7922	
72.12.209.146		 United States	Indiana	Lafayette	Wintek Corporation	11114	
85.203.44.80		 Netherlands	North Holland	Amsterdam	NForce Entertainment B.V.	43350	
95.222.167.189	ip-95-222-167-189.hsi15.unifymediagroup.de	 Germany	North Rhine-Westphalia	Bochum	Liberty Global B.V.	6830	

In conclusion, it does not seem difficult in practice to understand that a developed software, code is running on a sandbox system using the information obtained through WMI, therefore it is important to remember that it is possible for a malicious person or a member of a red team to benefit from this information and the IP addresses, IP blocks of sandbox systems to bypass sandbox analysis.

Hope to see you in the following articles.

Note: Those who are interested can download my presentation file titled "Sandbox Detection" which I discussed this topic in, from the following link, which was presented on November 22nd at the NOPcon International Hacker Conference.