MalBuster

Introduction

This room aims to be a practice room for Dissecting PE Headers and Static Analysis 1. In this scenario, you will act as one of the Reverse Engineers that will analyse malware samples based on the detections reported by your SOC team.

Prerequisites

This room requires basic knowledge of Malware Static Analysis. We recommend going through the following rooms before attempting this challenge.

- Intro to Malware Analysis
- Dissecting PE Headers
- Basic Static Analysis

Scenario

You are currently working as a Malware Reverse Engineer for your organization. Your team acts as a support for the SOC team when detections of unknown binaries occur. One of the SOC analysts triaged an alert triggered by binaries with unusual behaviour. Your task is to analyse the binaries detected by your SOC team and provide enough information to assist them in remediating the threat.

Investigation Platforms

The team has provided two investigation platforms, a FLARE VM and a REMnux VM. You may utilize the machines based on your preference.

If you prefer FLARE VM, you may start the machine attached to this task. Else, you may start the machine on the task below to start REMnux VM.

The machine will start in a split-screen view. In case the VM is not visible, use the blue Show Split View button at the top-right of the page.

You may also use the following credentials for alternative access via Remote Desktop (RDP):

- Username: administrator
- Password: letmein123!
- IP Address: MACHINE_IP

Lastly, you may find the malware samples on C:\Users\Administrator\Desktop\Samples.

WE ADVISE YOU NOT TO DOWNLOAD THE MALWARE SAMPLES TO YOUR HOST.

Challenge Questions

Investigation Platform

If you prefer REMnux, you may use the machine attached to this task by accessing it via the split-screen view.

Else, start the machine from the previous task to spin up the FLARE VM.

In addition, you can find the malware samples provided by the SOC team at /home/ubuntu/Desktop/Samples.

The machine will start in a split-screen view. In case the VM is not visible, use the blue Show Split View button at the top-right of the page.

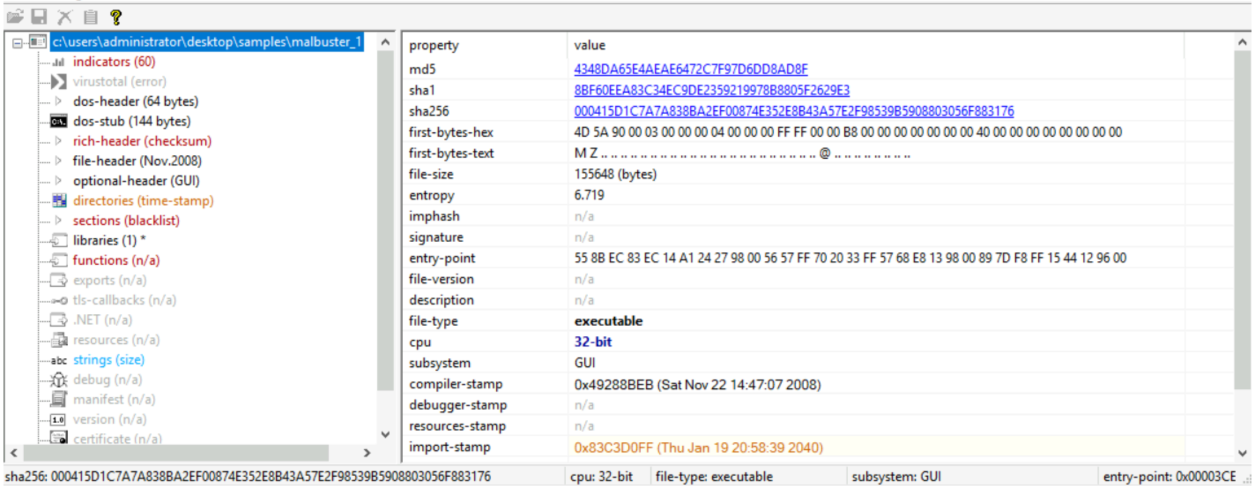
WE ADVISE YOU NOT TO DOWNLOAD THE MALWARE SAMPLES TO YOUR HOST.

Good luck!

****

Answer the questions below:

Based on the ARCHITECTURE of the binary, is malbuster_1 a 32-bit or a 64-bit application? (32-bit/64-bit)



c:\users\administrator\desktop\samples\malbuster_1
- indicators (60)
- virustotal (error)
- dos-header (64 bytes)
- dos-stub (144 bytes)
- rich-header (checksum)
- file-header (Nov.2008)
- optional-header (GUI)
- directories (time-stamp)
- sections (blacklist)
- libraries (1) *
- functions (n/a)
- exports (n/a)
- tls-callbacks (n/a)
- .NET (n/a)
- resources (n/a)
- strings (size)
- debug (n/a)
- manifest (n/a)
- version (n/a)
- certificate (n/a)

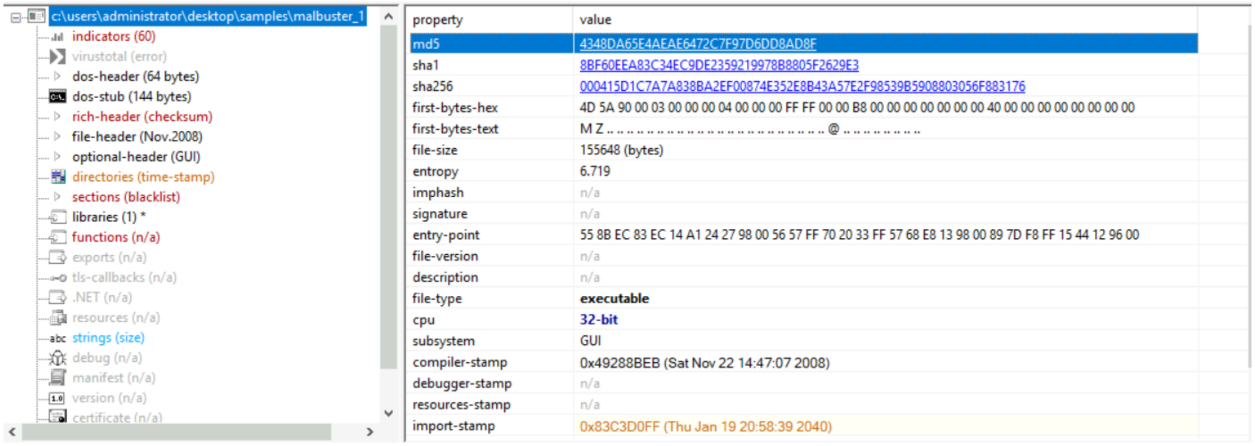
property	value
md5	4348DA65E4AEAE6472C7F97D6DD8AD8F
sha1	8BF60EEA83C34EC9DE2359219978B8805F2629E3
sha256	000415D1C7A7A838BA2EF00874E352E8B43A57E2F98539B5908803056F883176
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	155648 (bytes)
entropy	6.719
imphash	n/a
signature	n/a
entry-point	55 8B EC 83 EC 14 A1 24 27 98 00 56 57 FF 70 20 33 FF 57 68 E8 13 98 00 89 7D F8 FF 15 44 12 96 00
file-version	n/a
description	n/a
file-type	executable
cpu	32-bit
subsystem	GUI
compiler-stamp	0x49288BEB (Sat Nov 22 14:47:07 2008)
debugger-stamp	n/a
resources-stamp	n/a
import-stamp	0x83C3D0FF (Thu Jan 19 20:58:39 2040)

sha256: 000415D1C7A7A838BA2EF00874E352E8B43A57E2F98539B5908803056F883176 | cpu: 32-bit | file-type: executable | subsystem: GUI | entry-point: 0x00003CE

Opening sample 1 in PEStudio we can see, under the CPU section, this is a 32-bit application.

Answer: 32-bit

What is the MD5 hash of malbuster_1?

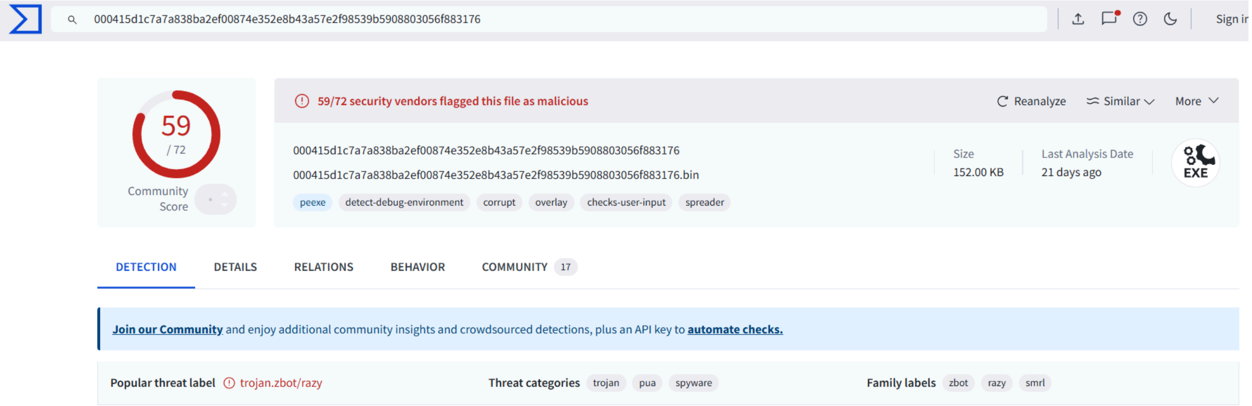


c:\users\administrator\desktop\samples\malbuster_1
- indicators (60)
- virustotal (error)
- dos-header (64 bytes)
- dos-stub (144 bytes)
- rich-header (checksum)
- file-header (Nov.2008)
- optional-header (GUI)
- directories (time-stamp)
- sections (blacklist)
- libraries (1) *
- functions (n/a)
- exports (n/a)
- tls-callbacks (n/a)
- .NET (n/a)
- resources (n/a)
- strings (size)
- debug (n/a)
- manifest (n/a)
- version (n/a)
- certificate (n/a)

property	value
md5	4348DA65E4AEAE6472C7F97D6DD8AD8F
sha1	8BF60EEA83C34EC9DE2359219978B8805F2629E3
sha256	000415D1C7A7A838BA2EF00874E352E8B43A57E2F98539B5908803056F883176
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	155648 (bytes)
entropy	6.719
imphash	n/a
signature	n/a
entry-point	55 8B EC 83 EC 14 A1 24 27 98 00 56 57 FF 70 20 33 FF 57 68 E8 13 98 00 89 7D F8 FF 15 44 12 96 00
file-version	n/a
description	n/a
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x49288BEB (Sat Nov 22 14:47:07 2008)
debugger-stamp	n/a
resources-stamp	n/a
import-stamp	0x83C3D0FF (Thu Jan 19 20:58:39 2040)

Answer: 4348da65e4aeae6472c7f97d6dd8ad8f

Using the hash, what is the popular threat label of malbuster_1 according to VirusTotal?



000415d1c7a7a838ba2ef00874e352e8b43a57e2f98539b5908803056f883176

59 / 72 Community Score

59/72 security vendors flagged this file as malicious

000415d1c7a7a838ba2ef00874e352e8b43a57e2f98539b5908803056f883176
000415d1c7a7a838ba2ef00874e352e8b43a57e2f98539b5908803056f883176.bin

peexe detect-debug-environment corrupt overlay checks-user-input spreader

Size 152.00 KB

Last Analysis Date 21 days ago

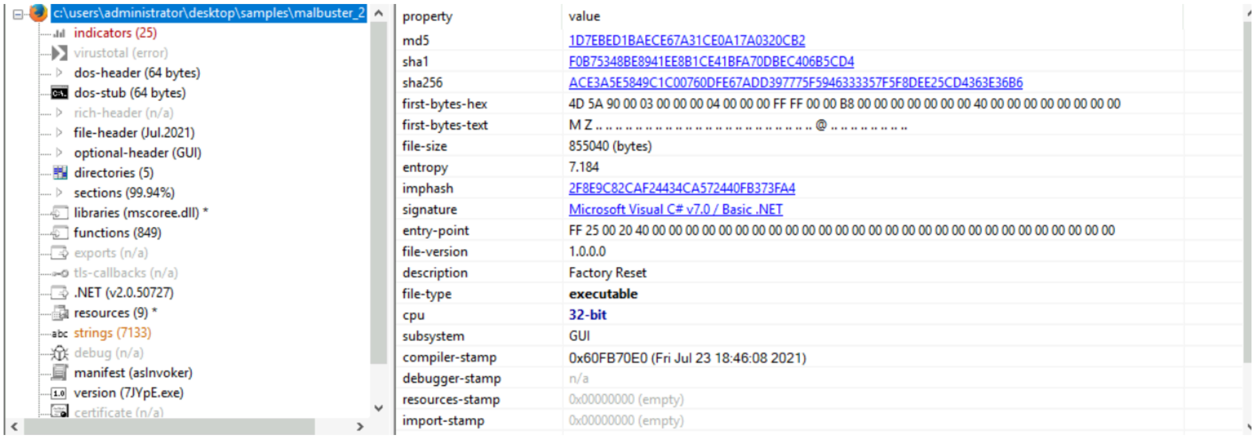
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.zbot/razy Threat categories trojan pua spyware Family labels zbot razy smrl

Answer: trojan.zbot/razy

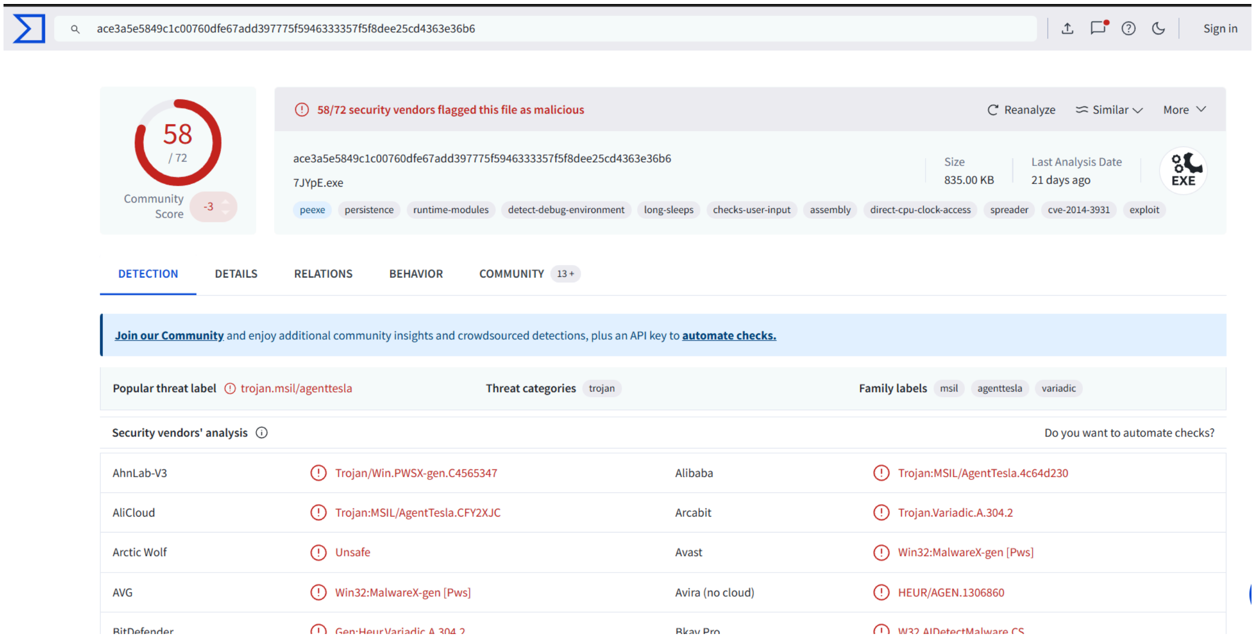
Based on VirusTotal detection, what is the malware signature of malbuster_2 according to Avira?

First, open sample 2 in PEStudio.



property	value
md5	1D7EBED1BAECE67A31CE0A17A0320CB2
sha1	F0B75348BE8941EE8B1CE41BFA70DBEC406B5CD4
sha256	ACE3A5E5849C1C00760DFE67ADD397775F5946333357F5F8DEE25CD4363E36B6
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	855040 (bytes)
entropy	7.184
imphash	2F8E9C82CAF24434CA572440FB373FA4
signature	Microsoft Visual C# v7.0 / Basic .NET
entry-point	FF 25 00 20 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
file-version	1.0.0.0
description	Factory Reset
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x60FB70E0 (Fri Jul 23 18:46:08 2021)
debugger-stamp	n/a
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)

Search the hash on VirusTotal.



ace3a5e5849c1c00760dfe67add397775f5946333357f5f8dee25cd4363e36b6

58 / 72 Community Score -3

58/72 security vendors flagged this file as malicious

ace3a5e5849c1c00760dfe67add397775f5946333357f5f8dee25cd4363e36b6
7JYpE.exe

Size 835.00 KB
Last Analysis Date 21 days ago

peexe persistence runtime-modules detect-debug-environment long-sleeps checks-user-input assembly direct-cpu-clock-access spreader cve-2014-3931 exploit

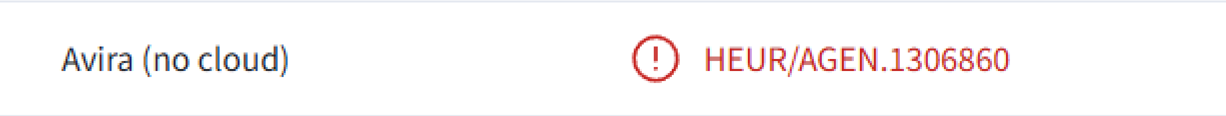
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.msil/agenttesla
Threat categories trojan
Family labels msil agenttesla variadic

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan/Win.PWSX-gen.C4565347	Alibaba	Trojan:MSIL/AgentTesla.4c64d230
AliCloud	Trojan:MSIL/AgentTesla.CFY2XJC	Arcabit	Trojan.Variadic.A.304.2
Arctic Wolf	Unsafe	Avast	Win32:MalwareX-gen [Pws]
AVG	Win32:MalwareX-gen [Pws]	Avira (no cloud)	HEUR/AGEN.1306860
BitDefender	Gen:Heur.Variadic.A.304.2	Bkav Pro	W32.AIDetectMalware.CS

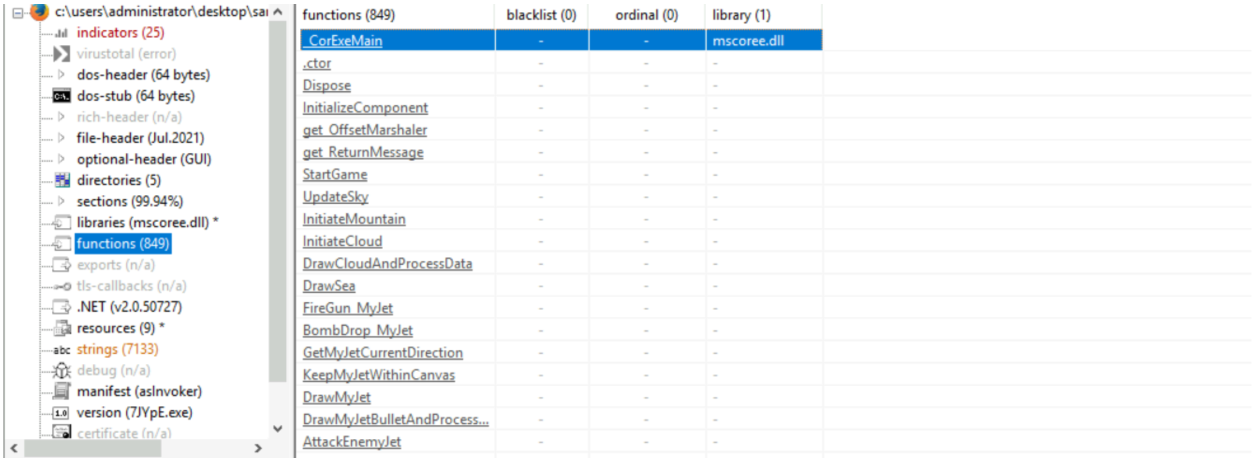


Avira (no cloud)	HEUR/AGEN.1306860

Answer: HEUR/AGEN.1306860

malbuster_2 imports the function _CorExeMain. From which DLL file does it import this function?

Open the functions tab on PEStudio.



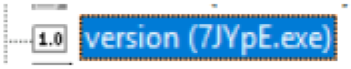
- c:\users\administrator\desktop\sa
- indicators (25)
- virustotal (error)
- dos-header (64 bytes)
- dos-stub (64 bytes)
- rich-header (n/a)
- file-header (Jul.2021)
- optional-header (GUI)
- directories (5)
- sections (99.94%)
- libraries (mscoree.dll) *
- functions (849)
- exports (n/a)
- tls-callbacks (n/a)
- .NET (v2.0.50727)
- resources (9) *
- strings (7133)
- debug (n/a)
- manifest (asInvoker)
- version (7JYpE.exe)
- certificate (n/a)

functions (849)	blacklist (0)	ordinal (0)	library (1)
_CorExeMain	-	-	mscoree.dll
.ctor	-	-	-
Dispose	-	-	-
InitializeComponent	-	-	-
get_OffsetMarshaler	-	-	-
get_ReturnMessage	-	-	-
StartGame	-	-	-
UpdateSky	-	-	-
InitiateMountain	-	-	-
InitiateCloud	-	-	-
DrawCloudAndProcessData	-	-	-
DrawSea	-	-	-
FireGun_MyJet	-	-	-
BombDrop_MyJet	-	-	-
GetMyJetCurrentDirection	-	-	-
KeepMyJetWithinCanvas	-	-	-
DrawMyJet	-	-	-
DrawMyJetBulletAndProcess...	-	-	-
AttackEnemyJet	-	-	-

And _CorExeMain is the first function on the list and to the right under the library section we see the DLL that imports the function.

Answer: mscoree.dll

Based on the VS_VERSION_INFO header, what is the original name of malbuster_2?

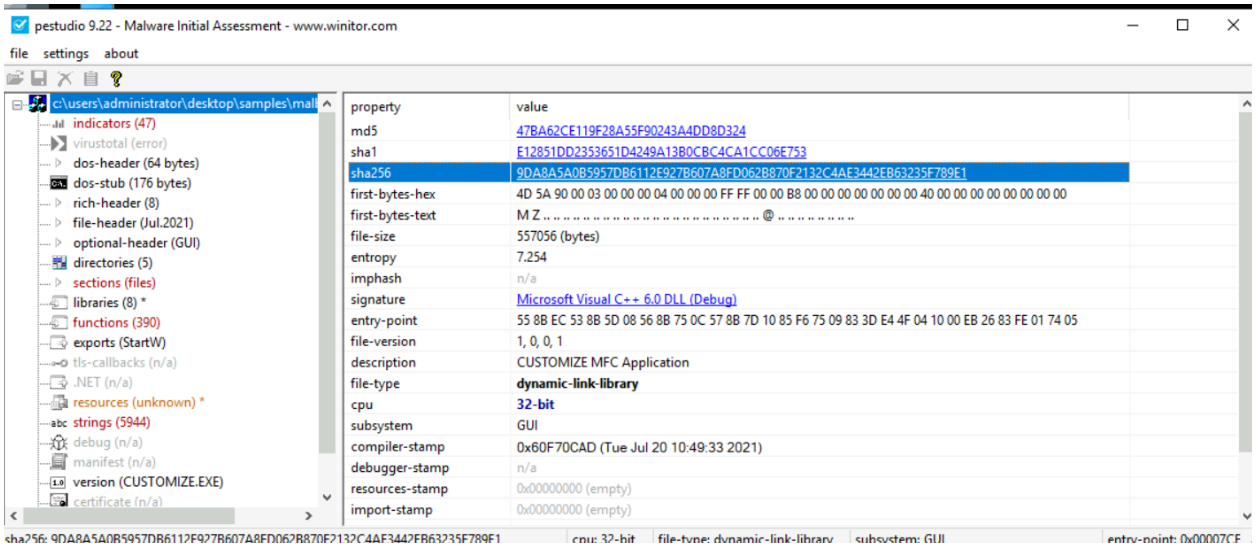


version (7JYpE.exe)

Answer: 7JYpE.exe

Using the hash of malbuster_3, what is its malware signature based on abuse.ch?

Again, open sample 3 in PEStudio but this time we'll copy the SHA256 hash and take it over to bazaar.abuse.ch to search for the sample.



pestudio 9.22 - Malware Initial Assessment - www.winitor.com
file settings about

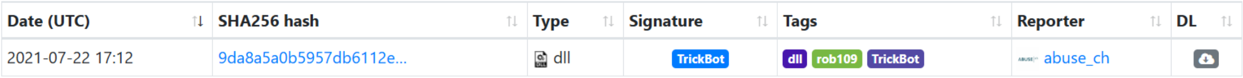
c:\users\administrator\desktop\samples\mal

- indicators (47)
- virustotal (error)
- dos-header (64 bytes)
- dos-stub (176 bytes)
- rich-header (8)
- file-header (Jul.2021)
- optional-header (GUI)
- directories (5)
- sections (files)
- libraries (8) *
- functions (390)
- exports (StartW)
- tls-callbacks (n/a)
- .NET (n/a)
- resources (unknown) *
- strings (5944)
- debug (n/a)
- manifest (n/a)
- version (CUSTOMIZE.EXE)
- certificate (n/a)

property	value
md5	47BA62CE119F28A55F90243A4DD8D324
sha1	E12851DD2353651D4249A13B0CBC4CA1CC06E753
sha256	9DA8A5A0B5957DB6112E927B607A8FD062B870F2132C4AE3442EB63235F789E1
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	557056 (bytes)
entropy	7.254
imphash	n/a
signature	Microsoft Visual C++ 6.0 DLL (Debug)
entry-point	55 8B EC 53 8B 5D 08 56 8B 75 0C 57 8B 7D 10 85 F6 75 09 83 3D E4 4F 04 10 00 EB 26 83 FE 01 74 05
file-version	1, 0, 0, 1
description	CUSTOMIZE MFC Application
file-type	**dynamic-link-library**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x60F70CAD (Tue Jul 20 10:49:33 2021)
debugger-stamp	n/a
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)

sha256: 9DA8A5A0B5957DB6112E927B607A8FD062B870F2132C4AE3442EB63235F789E1 | cpu: 32-bit | file-type: dynamic-link-library | subsystem: GUI | entry-point: 0x00007CE

Search sha256:9DA8A5A0B5957DB6112E927B607A8FD062B870F2132C4AE3442EB63235F789E1 on the Bazaar.



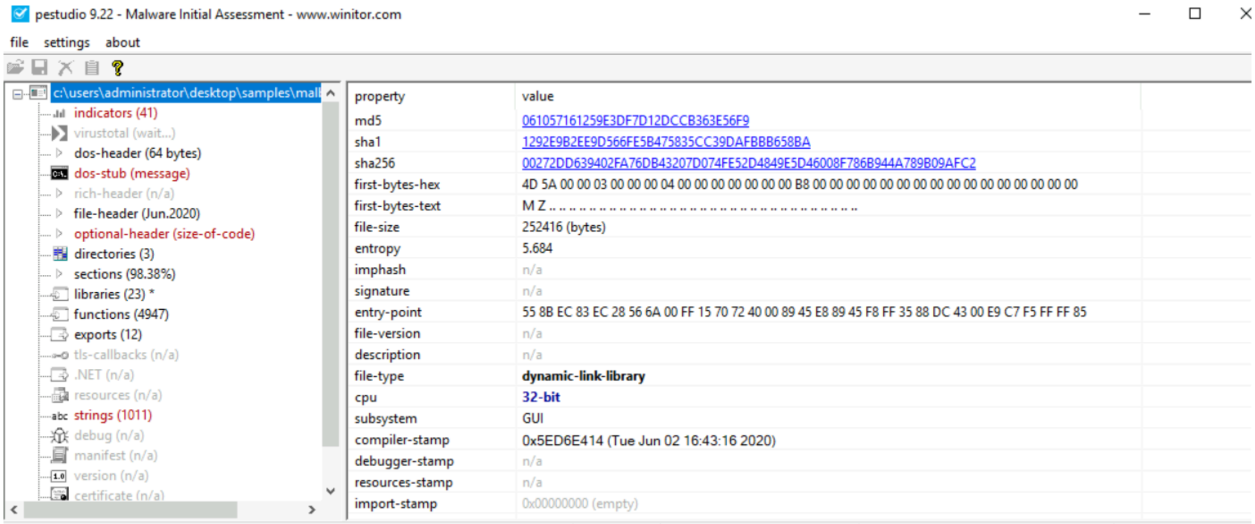
Date (UTC)	SHA256 hash	Type	Signature	Tags	Reporter	DL
2021-07-22 17:12	9da8a5a0b5957db6112e...	dll	TrickBot	dll rob109 TrickBot	abuse_ch	

We can see the signature is tagged Trickbot.

Answer: Trickbot

Using the hash of malbuster_4, what is its malware signature based on abuse.ch?

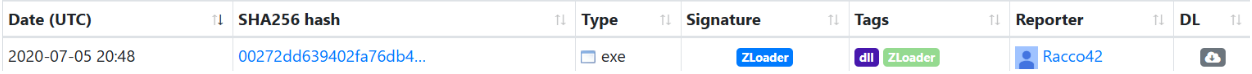
Repeat the process of the last question but this time using sample 4.



property	value
md5	061057161259E3DF7D12DCCB363E56F9
sha1	1292E9B2EE9D566FE5B475835CC39DAFBBB658BA
sha256	00272DD639402FA76DB43207D074FE52D4849E5D46008F786B944A789B09AFC2
first-bytes-hex	4D 5A 00 00 03 00 00 00 04 00 00 00 00 00 00 00 B8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
first-bytes-text	M Z .. ..
file-size	252416 (bytes)
entropy	5.684
imphash	n/a
signature	n/a
entry-point	55 8B EC 83 EC 28 56 6A 00 FF 15 70 72 40 00 89 45 E8 89 45 F8 FF 35 88 DC 43 00 E9 C7 F5 FF FF 85
file-version	n/a
description	n/a
file-type	**dynamic-link-library**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x5ED6E414 (Tue Jun 02 16:43:16 2020)
debugger-stamp	n/a
resources-stamp	n/a
import-stamp	0x00000000 (empty)

Open the file in PEStudio and copy the SHA256 hash.

Search sha256:00272DD639402FA76DB43207D074FE52D4849E5D46008F786B944A789B09AFC2 on bazaar.abuse.ch.

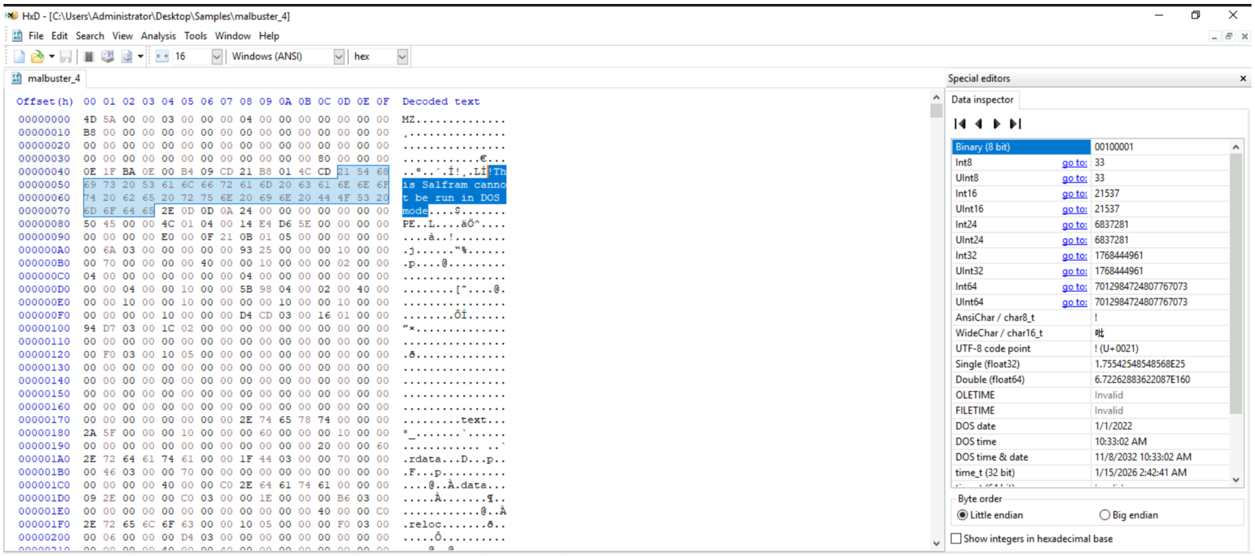


Date (UTC)	SHA256 hash	Type	Signature	Tags	Reporter	DL
2020-07-05 20:48	00272dd639402fa76db4...	exe	ZLoader	dll ZLoader	Racco42	

Signature is tagged as ZLoader.

Answer: ZLoader

What is the message found in the DOS_STUB of malbuster_4?



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00000000  4D 5A 00 00 03 00 00 00 04 00 00 00 00 00 00 00  MZ..............
00000010  B8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ¸...............
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00  ............€...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  69 73 20 53 61 6C 66 72 61 6D 20 63 61 6E 6E 6F  is Salfram canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  50 45 00 00 4C 01 04 00 14 E4 D6 5E 00 00 00 00  PE..L....äÖ^....
00000090  00 00 00 00 E0 00 0F 21 0B 01 05 00 00 00 00 00  ....à..!........
000000A0  00 6A 03 00 00 00 00 00 93 25 00 00 00 10 00 00  .j......“%......
000000B0  00 70 00 00 00 00 40 00 00 10 00 00 00 02 00 00  .p....@.........
000000C0  04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00  ................
000000D0  00 00 04 00 00 10 00 00 5B 98 04 00 02 00 40 00  ........[˜....@.
000000E0  00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00  ................
000000F0  00 00 00 00 10 00 00 00 D4 CD 03 00 16 01 00 00  ........ÔÍ......
00000100  94 D7 03 00 1C 02 00 00 00 00 00 00 00 00 00 00  ”×..............
00000110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000120  00 F0 03 00 10 05 00 00 00 00 00 00 00 00 00 00  .ð..............
00000130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000170  00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00  .........text...
00000180  2A 5F 00 00 00 10 00 00 00 60 00 00 00 10 00 00  *_.......`......
00000190  00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60  ............ ..`
000001A0  2E 72 64 61 74 61 00 00 1F 44 03 00 00 70 00 00  .rdata...D...p..
000001B0  00 46 03 00 00 70 00 00 00 00 00 00 00 00 00 00  .F...p..........
000001C0  00 00 00 00 40 00 00 C0 2E 64 61 74 61 00 00 00  ....@..À.data...
000001D0  09 2E 00 00 00 C0 03 00 00 1E 00 00 00 B6 03 00  .....À.......¶..
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 C0  ............@..À
000001F0  2E 72 65 6C 6F 63 00 00 10 05 00 00 00 F0 03 00  .reloc.......ð..
00000200  00 06 00 00 00 D4 03 00 00 00 00 00 00 00 00 00  .....Ô..........
```

Binary (8 bit)		00100001
Int8	go to:	33
UInt8	go to:	33
Int16	go to:	21537
UInt16	go to:	21537
Int24	go to:	6837281
UInt24	go to:	6837281
Int32	go to:	1768444961
UInt32	go to:	1768444961
Int64	go to:	7012984724807767073
UInt64	go to:	7012984724807767073
AnsiChar / char8_t		!
WideChar / char16_t		吡
UTF-8 code point		! (U+0021)
Single (float32)		1.75542548548568E25
Double (float64)		6.72262883622087E160
OLETIME		Invalid
FILETIME		Invalid
DOS date		1/1/2022
DOS time		10:33:02 AM
DOS time & date		11/8/2032 10:33:02 AM
time_t (32 bit)		1/15/2026 2:42:41 AM

Byte order

● Little endian ○ Big endian

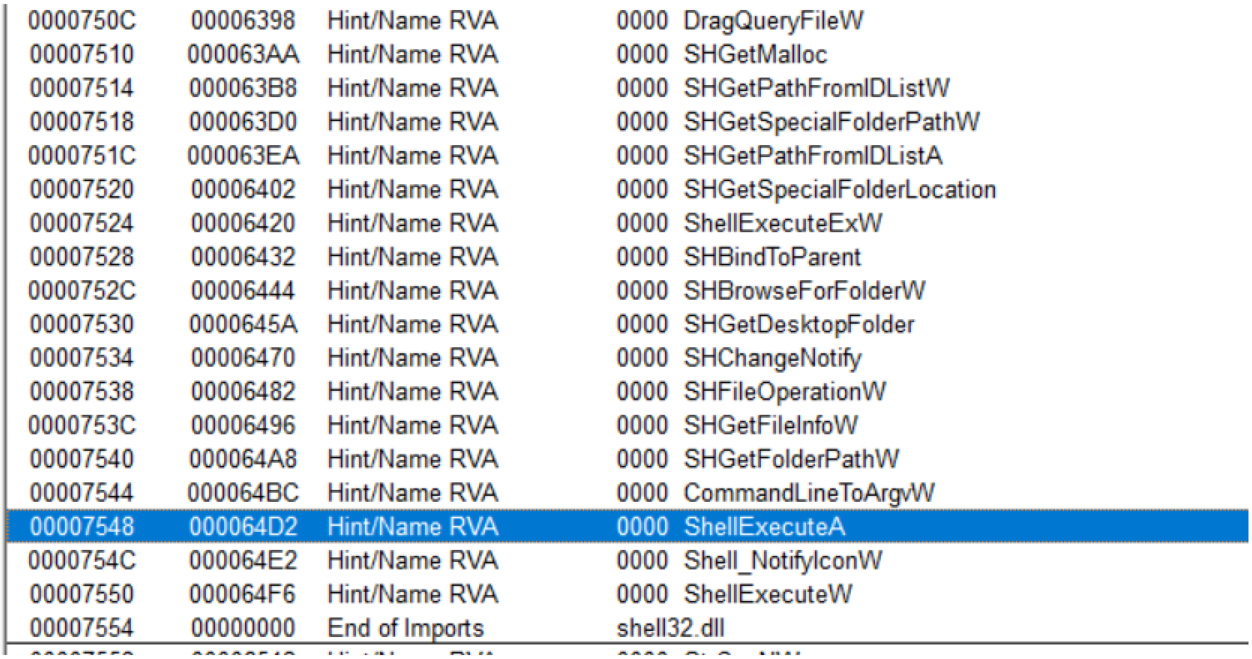
□ Show integers in hexadecimal base

Using HxD we can open sample 4 and see the message.

Answer: !This Salfram cannot be run in DOS mode.

malbuster_4 imports the function ShellExecuteA. From which DLL file does it import this function?

Like an earlier question I went to PEStudio looking under functions to find ShellExecuteA and the DLL that imports it but when I did so I only got gibberish. So I switched to trying PEView.

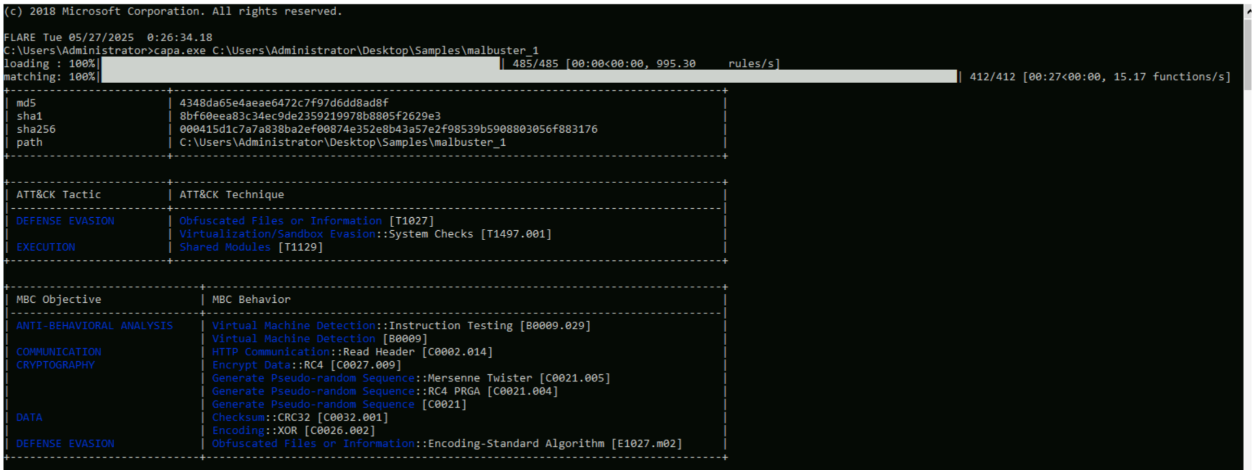


0000750C	00006398	Hint/Name RVA	0000 DragQueryFileW
00007510	000063AA	Hint/Name RVA	0000 SHGetMalloc
00007514	000063B8	Hint/Name RVA	0000 SHGetPathFromIDListW
00007518	000063D0	Hint/Name RVA	0000 SHGetSpecialFolderPathW
0000751C	000063EA	Hint/Name RVA	0000 SHGetPathFromIDListA
00007520	00006402	Hint/Name RVA	0000 SHGetSpecialFolderLocation
00007524	00006420	Hint/Name RVA	0000 ShellExecuteExW
00007528	00006432	Hint/Name RVA	0000 SHBindToParent
0000752C	00006444	Hint/Name RVA	0000 SHBrowseForFolderW
00007530	0000645A	Hint/Name RVA	0000 SHGetDesktopFolder
00007534	00006470	Hint/Name RVA	0000 SHChangeNotify
00007538	00006482	Hint/Name RVA	0000 SHFileOperationW
0000753C	00006496	Hint/Name RVA	0000 SHGetFileInfoW
00007540	000064A8	Hint/Name RVA	0000 SHGetFolderPathW
00007544	000064BC	Hint/Name RVA	0000 CommandLineToArgvW
00007548	000064D2	Hint/Name RVA	0000 ShellExecuteA
0000754C	000064E2	Hint/Name RVA	0000 Shell_NotifyIconW
00007550	000064F6	Hint/Name RVA	0000 ShellExecuteW
00007554	00000000	End of Imports	shell32.dll

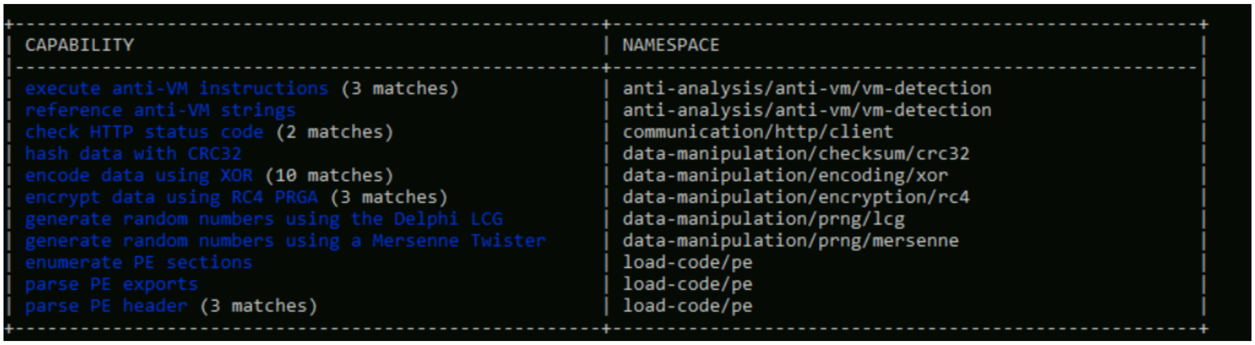
Here we see all the imports from shell32.dll and the ShellExecuteA we're looking for is here.

Answer: shell32.dll

Using capa, how many anti-VM instructions were identified in malbuster_1?



```
(c) 2018 Microsoft Corporation. All rights reserved.

FLARE Tue 05/27/2025  0:26:34.18
C:\Users\Administrator>capa.exe C:\Users\Administrator\Desktop\Samples\malbuster_1
loading : 100%|                                                    | 485/485 [00:00<00:00, 995.30     rules/s]
matching: 100%|                                                                                                  | 412/412 [00:27<00:00, 15.17 functions/s]
+------------------------+-------------------------------------------------------------------------------------+
| md5                    | 4348da65e4aeae6472c7f97d6dd8ad8f                                                    |
| sha1                   | 8bf60eea83c34ec9de2359219978b8805f2629e3                                            |
| sha256                 | 000415d1c7a7a838ba2ef00874e352e8b43a57e2f98539b5908803056f883176                    |
| path                   | C:\Users\Administrator\Desktop\Samples\malbuster_1                                  |
+------------------------+-------------------------------------------------------------------------------------+

+------------------------+-------------------------------------------------------------------------------------+
| ATT&CK Tactic          | ATT&CK Technique                                                                    |
|------------------------+-------------------------------------------------------------------------------------|
| DEFENSE EVASION        | Obfuscated Files or Information [T1027]                                             |
|                        | Virtualization/Sandbox Evasion::System Checks [T1497.001]                           |
| EXECUTION              | Shared Modules [T1129]                                                              |
+------------------------+-------------------------------------------------------------------------------------+

+-----------------------------+--------------------------------------------------------------------------------+
| MBC Objective               | MBC Behavior                                                                   |
|-----------------------------+--------------------------------------------------------------------------------|
| ANTI-BEHAVIORAL ANALYSIS    | Virtual Machine Detection::Instruction Testing [B0009.029]                     |
|                             | Virtual Machine Detection [B0009]                                              |
| COMMUNICATION               | HTTP Communication::Read Header [C0002.014]                                    |
| CRYPTOGRAPHY                | Encrypt Data::RC4 [C0027.009]                                                  |
|                             | Generate Pseudo-random Sequence::Mersenne Twister [C0021.005]                  |
|                             | Generate Pseudo-random Sequence::RC4 PRGA [C0021.004]                          |
|                             | Generate Pseudo-random Sequence [C0021]                                        |
| DATA                        | Checksum::CRC32 [C0032.001]                                                    |
|                             | Encoding::XOR [C0026.002]                                                      |
| DEFENSE EVASION             | Obfuscated Files or Information::Encoding-Standard Algorithm [E1027.m02]       |
+-----------------------------+--------------------------------------------------------------------------------+
```

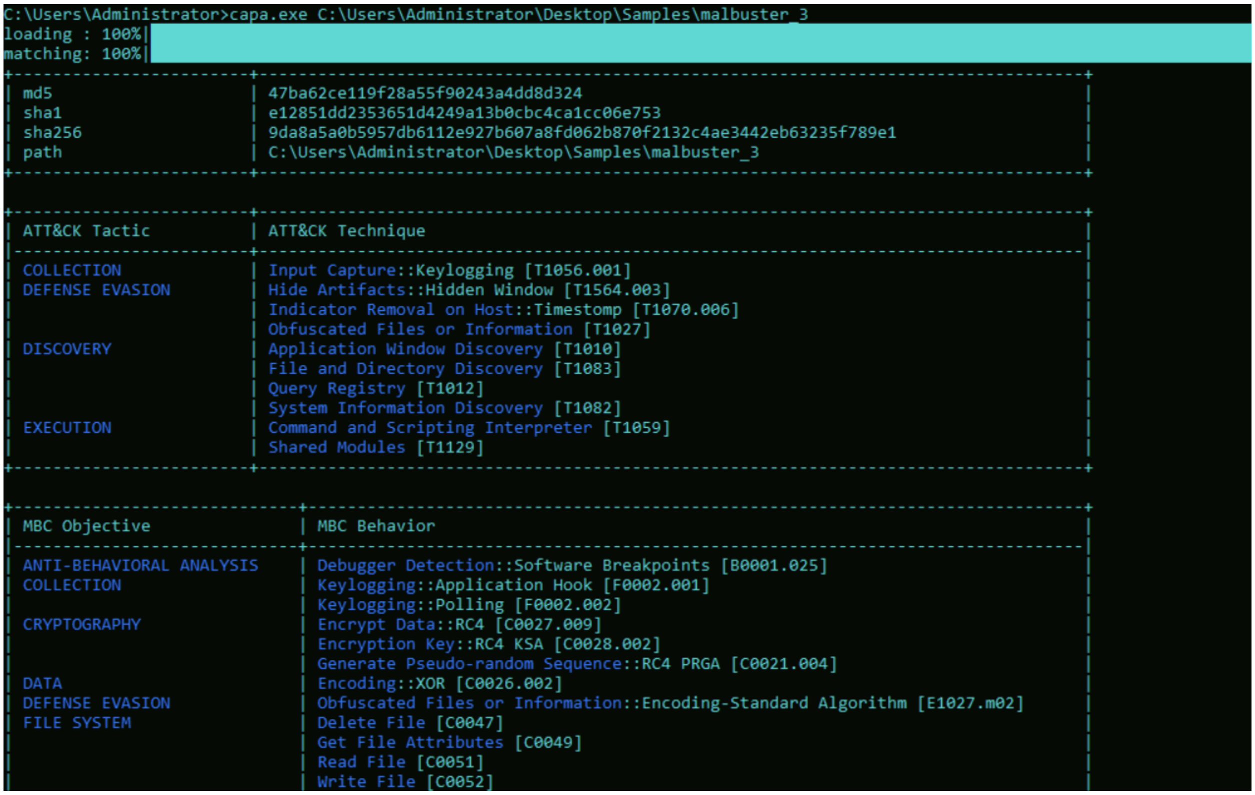


```
+-----------------------------------------------------+------------------------------------------------------+
| CAPABILITY                                          | NAMESPACE                                            |
|-----------------------------------------------------+------------------------------------------------------|
| execute anti-VM instructions (3 matches)            | anti-analysis/anti-vm/vm-detection                   |
| reference anti-VM strings                           | anti-analysis/anti-vm/vm-detection                   |
| check HTTP status code (2 matches)                  | communication/http/client                            |
| hash data with CRC32                                | data-manipulation/checksum/crc32                     |
| encode data using XOR (10 matches)                  | data-manipulation/encoding/xor                       |
| encrypt data using RC4 PRGA (3 matches)             | data-manipulation/encryption/rc4                     |
| generate random numbers using the Delphi LCG        | data-manipulation/prng/lcg                           |
| generate random numbers using a Mersenne Twister    | data-manipulation/prng/mersenne                      |
| enumerate PE sections                               | load-code/pe                                         |
| parse PE exports                                    | load-code/pe                                         |
| parse PE header (3 matches)                         | load-code/pe                                         |
+-----------------------------------------------------+------------------------------------------------------+
```

Answer: 3

Using capa, which binary can log keystrokes?

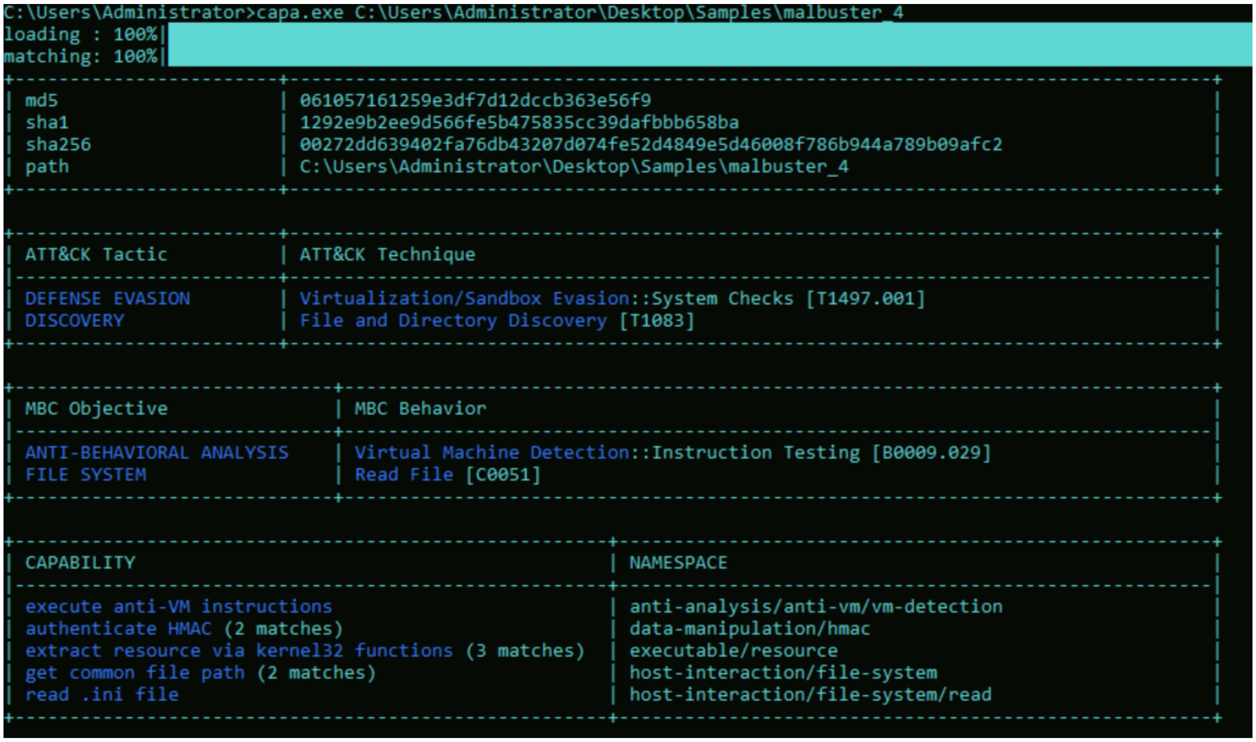
Running Capa on sample 3 we see that it has keylogging capabilities.



```
C:\Users\Administrator>capa.exe C:\Users\Administrator\Desktop\Samples\malbuster_3
loading : 100%|
matching: 100%|
+------------------------+------------------------------------------------------------------------------------+
| md5                    | 47ba62ce119f28a55f90243a4dd8d324                                                   |
| sha1                   | e12851dd2353651d4249a13b0cbc4ca1cc06e753                                           |
| sha256                 | 9da8a5a0b5957db6112e927b607a8fd062b870f2132c4ae3442eb63235f789e1                   |
| path                   | C:\Users\Administrator\Desktop\Samples\malbuster_3                                 |
+------------------------+------------------------------------------------------------------------------------+

+------------------------+------------------------------------------------------------------------------------+
| ATT&CK Tactic          | ATT&CK Technique                                                                   |
|------------------------+------------------------------------------------------------------------------------|
| COLLECTION             | Input Capture::Keylogging [T1056.001]                                              |
| DEFENSE EVASION        | Hide Artifacts::Hidden Window [T1564.003]                                          |
|                        | Indicator Removal on Host::Timestomp [T1070.006]                                   |
|                        | Obfuscated Files or Information [T1027]                                            |
| DISCOVERY              | Application Window Discovery [T1010]                                               |
|                        | File and Directory Discovery [T1083]                                               |
|                        | Query Registry [T1012]                                                             |
|                        | System Information Discovery [T1082]                                               |
| EXECUTION              | Command and Scripting Interpreter [T1059]                                          |
|                        | Shared Modules [T1129]                                                             |
+------------------------+------------------------------------------------------------------------------------+

+-----------------------------+-------------------------------------------------------------------------------+
| MBC Objective               | MBC Behavior                                                                  |
|-----------------------------+-------------------------------------------------------------------------------|
| ANTI-BEHAVIORAL ANALYSIS    | Debugger Detection::Software Breakpoints [B0001.025]                          |
| COLLECTION                  | Keylogging::Application Hook [F0002.001]                                      |
|                             | Keylogging::Polling [F0002.002]                                               |
| CRYPTOGRAPHY                | Encrypt Data::RC4 [C0027.009]                                                 |
|                             | Encryption Key::RC4 KSA [C0028.002]                                           |
|                             | Generate Pseudo-random Sequence::RC4 PRGA [C0021.004]                         |
| DATA                        | Encoding::XOR [C0026.002]                                                     |
| DEFENSE EVASION             | Obfuscated Files or Information::Encoding-Standard Algorithm [E1027.m02]      |
| FILE SYSTEM                 | Delete File [C0047]                                                           |
|                             | Get File Attributes [C0049]                                                   |
|                             | Read File [C0051]                                                             |
|                             | Write File [C0052]                                                            |
```

Answer: malbuster_3

Using capa, what is the MITRE ID of the DISCOVERY technique used by malbuster_4?

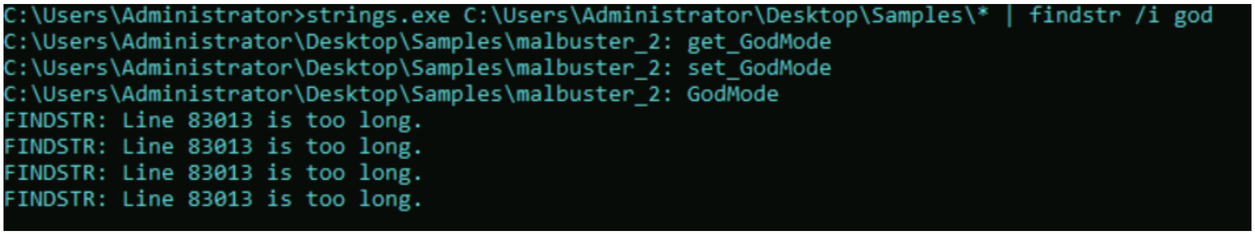


```
C:\Users\Administrator>capa.exe C:\Users\Administrator\Desktop\Samples\malbuster_4
loading : 100%|
matching: 100%|
+------------------------+------------------------------------------------------------------------------------+
| md5                    | 061057161259e3df7d12dccb363e56f9                                                   |
| sha1                   | 1292e9b2ee9d566fe5b475835cc39dafbbb658ba                                           |
| sha256                 | 00272dd639402fa76db43207d074fe52d4849e5d46008f786b944a789b09afc2                   |
| path                   | C:\Users\Administrator\Desktop\Samples\malbuster_4                                 |
+------------------------+------------------------------------------------------------------------------------+

+------------------------+------------------------------------------------------------------------------------+
| ATT&CK Tactic          | ATT&CK Technique                                                                   |
|------------------------+------------------------------------------------------------------------------------|
| DEFENSE EVASION        | Virtualization/Sandbox Evasion::System Checks [T1497.001]                          |
| DISCOVERY              | File and Directory Discovery [T1083]                                               |
+------------------------+------------------------------------------------------------------------------------+

+-----------------------------+-------------------------------------------------------------------------------+
| MBC Objective               | MBC Behavior                                                                  |
|-----------------------------+-------------------------------------------------------------------------------|
| ANTI-BEHAVIORAL ANALYSIS    | Virtual Machine Detection::Instruction Testing [B0009.029]                    |
| FILE SYSTEM                 | Read File [C0051]                                                             |
+-----------------------------+-------------------------------------------------------------------------------+

+------------------------------------------------------+-------------------------------------------------------+
| CAPABILITY                                           | NAMESPACE                                             |
|------------------------------------------------------+-------------------------------------------------------|
| execute anti-VM instructions                         | anti-analysis/anti-vm/vm-detection                    |
| authenticate HMAC (2 matches)                        | data-manipulation/hmac                                |
| extract resource via kernel32 functions (3 matches)  | executable/resource                                   |
| get common file path (2 matches)                     | host-interaction/file-system                          |
| read .ini file                                       | host-interaction/file-system/read                     |
+------------------------------------------------------+-------------------------------------------------------+
```

Answer: T1083

Which binary contains the string GodMode?

Using the command *strings.exe C:\Users\Administrator\Desktop\Samples\* | *findstr /i god* to search all the samples for strings that contain “god” regardless of capitalization I got the answer.

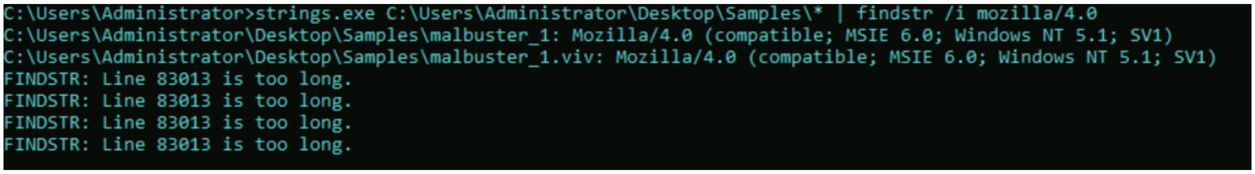


```
C:\Users\Administrator>strings.exe C:\Users\Administrator\Desktop\Samples\* | findstr /i god
C:\Users\Administrator\Desktop\Samples\malbuster_2: get_GodMode
C:\Users\Administrator\Desktop\Samples\malbuster_2: set_GodMode
C:\Users\Administrator\Desktop\Samples\malbuster_2: GodMode
FINDSTR: Line 83013 is too long.
FINDSTR: Line 83013 is too long.
FINDSTR: Line 83013 is too long.
FINDSTR: Line 83013 is too long.
```

Answer: malbuster_2

Which binary contains the string Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)?

Repeat the same process as the previous question but search for Mozilla/4.0 instead. Command executed: *strings.exe C:\Users\Administrator\Desktop\Samples\* | *findstr /i mozilla/4.0*



```
C:\Users\Administrator>strings.exe C:\Users\Administrator\Desktop\Samples\* | findstr /i mozilla/4.0
C:\Users\Administrator\Desktop\Samples\malbuster_1: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
C:\Users\Administrator\Desktop\Samples\malbuster_1.viv: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
FINDSTR: Line 83013 is too long.
FINDSTR: Line 83013 is too long.
FINDSTR: Line 83013 is too long.
FINDSTR: Line 83013 is too long.
```

Answer: malbuster_1