Name:Ramya Ajay

Roll No:CB.EN.P2CYS22004

MAL: Malware Introductory

Understanding Malware Campaigns (Task 2)

Two types : Targeted and Mass Campaign

Targeted

A "Targeted" attack is just that - targeted. In most cases, malware attacks that occur this way are created for a specific purpose against a specific target. A great example of this type of purpose could be the DarkHotel malware, whom is designed to steal information such as authentication details from government officials.

Mass Campaign

Mass Campaign classification can be akin to many real life examples, and is the most common type of attacks. The entire purpose of this type of Malware is to infect as many devices as possible and perform whatever it may - regardless of target.

What is the famous example of a targeted attack-esque Malware that targeted Iran?

```
Stuxnet
```

What is the name of the Ransomware that used the Eternalblue exploit in a "Mass Campaign" attack?

```
Wannacry
```

Identifying if a Malware Attack has Happened (Task 3)

Process of a malware attack

- Delivery
- Execution
- Maintaining Persistence

- Persistence
- Propagation

Two categories of fingerprints : Host Based and Network Based.

Name the first essential step of a Malware Attack?

```
Delivery
```

Now name the second essential step of a Malware Attack?

```
Execution
```

What type of signature is used to classify remnants of infection on a host?

```
Host-based signatures
```

What is the name of the other classification of signature used after a Malware attack?

```
Network-based signatures
```

Static Vs. Dynamic Analysis (Task 4)

Static Analysis

"Static Analysis" is used to gain a high-level abstraction of the sample - it can be fairly simple to decide if a piece of code is "malicious" or not with this method alone. At its core, this method is of the analysis of the sample at the state it presents itself as, without executing the code. Employing the use of techniques such as signature analysis via checksums means quick, efficient (albeit extremely brief) and safe analysis of malware.

Dynamic Analysis

"Dynamic Analysis" essentially involves executing the sample and observing what happens.It is the testing and evaluation of an application by examining the code without executing the application.

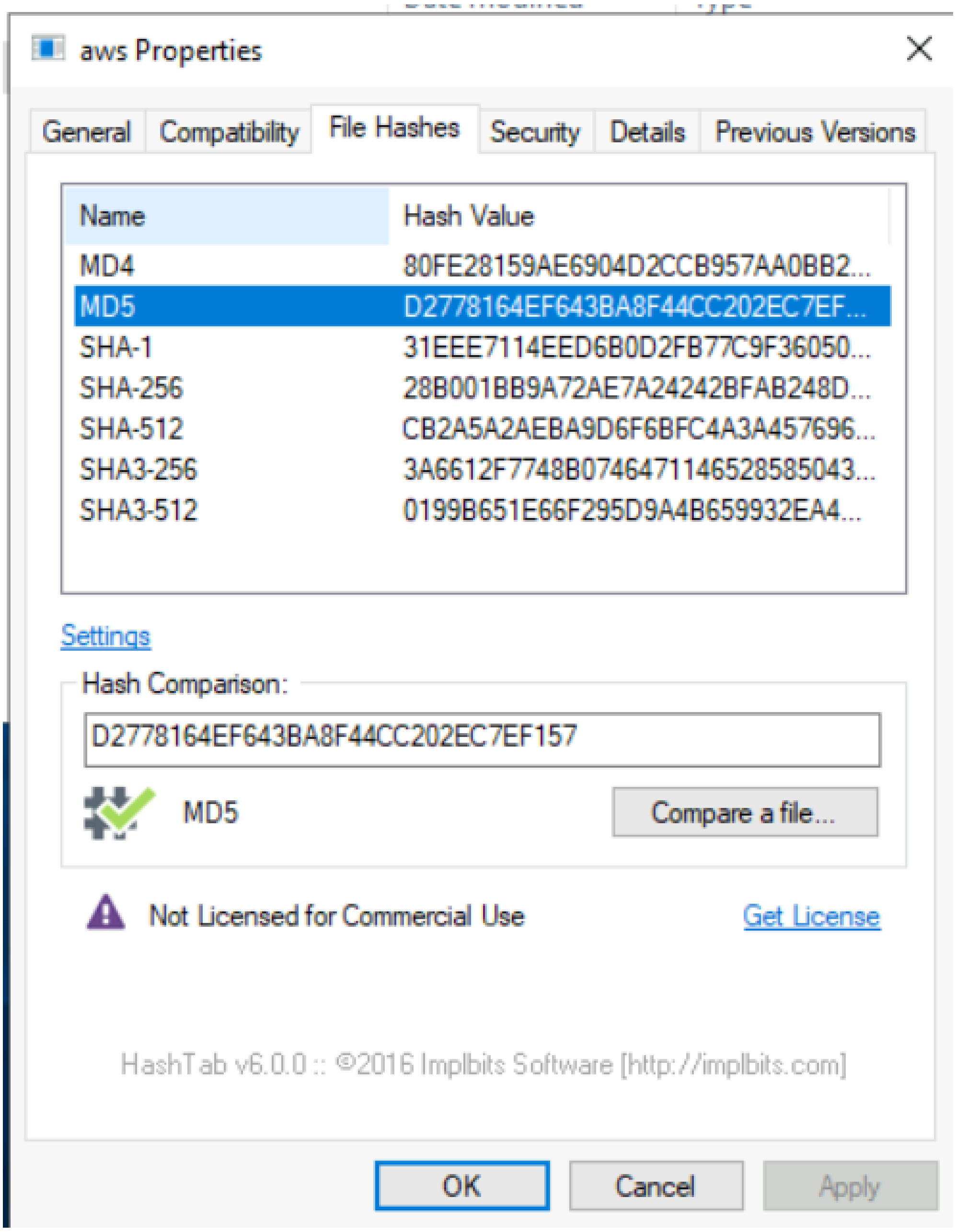
Obtaining MD5 Checksums of Provided Files (Task 7)

MD5 "Checksums" are a prominent attribute in the malware Community. Because there can be many variants of a family of Ransomware, these MD5 "Checksums" are cryptographic "fingerprints" of the files. This allows a uniformed identification throughout the community - especially with automated Sandboxes.

Identify the MD5 Checksums of the three files provided in "Task 7"

The MD5 Checksum of aws.exe

```
D2778164EF643BA8F44CC202EC7EF157
```



Name	Hash Value
MD4	80FE28159AE6904D2CCB957AA0BB2...
MD5	D2778164EF643BA8F44CC202EC7EF...
SHA-1	31EEE7114EED6B0D2FB77C9F36050...
SHA-256	28B001BB9A72AE7A24242BFAB248D...
SHA-512	CB2A5A2AEBA9D6F6BFC4A3A457696...
SHA3-256	3A6612F7748B0746471146528585043...
SHA3-512	0199B651E66F295D9A4B659932EA4...

Hash Comparison: D2778164EF643BA8F44CC202EC7EF157

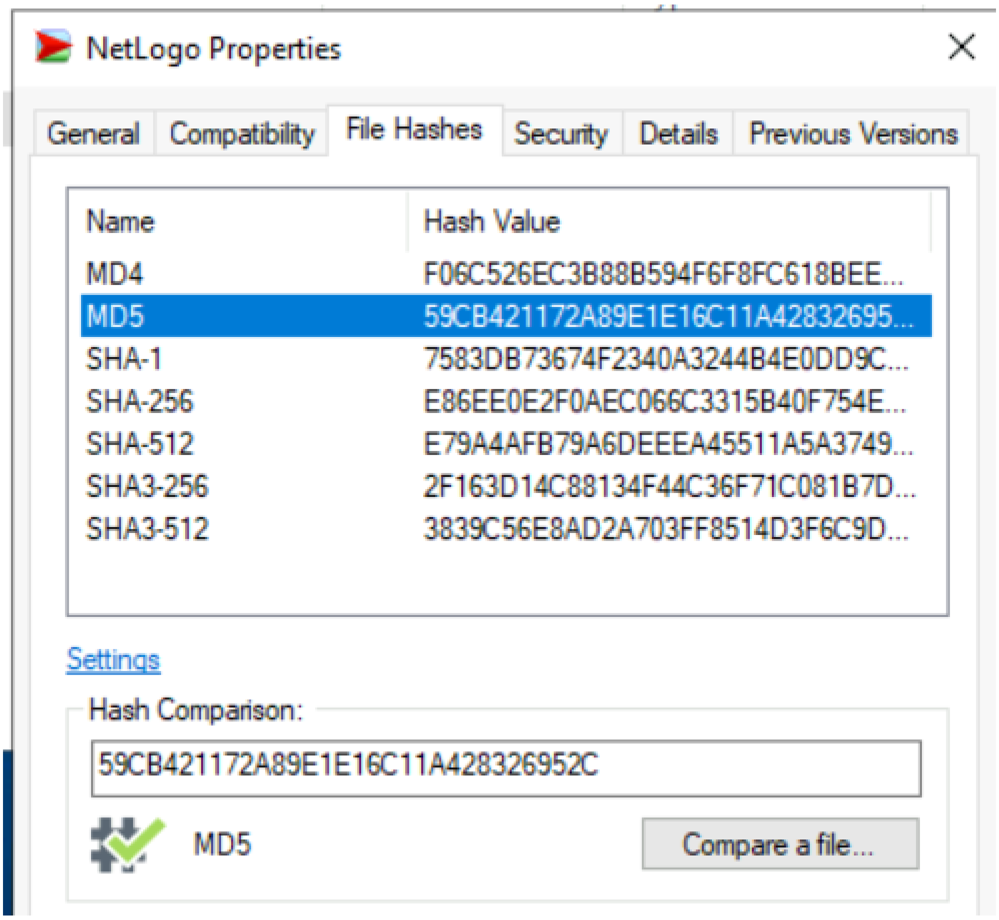
MD5

Not Licensed for Commercial Use

HashTab v6.0.0 :: ©2016 Implbits Software [http://implbits.com]

The MD5 Checksum of Netlogo.exe

```
59CB421172A89E1E16C11A428326952C
```



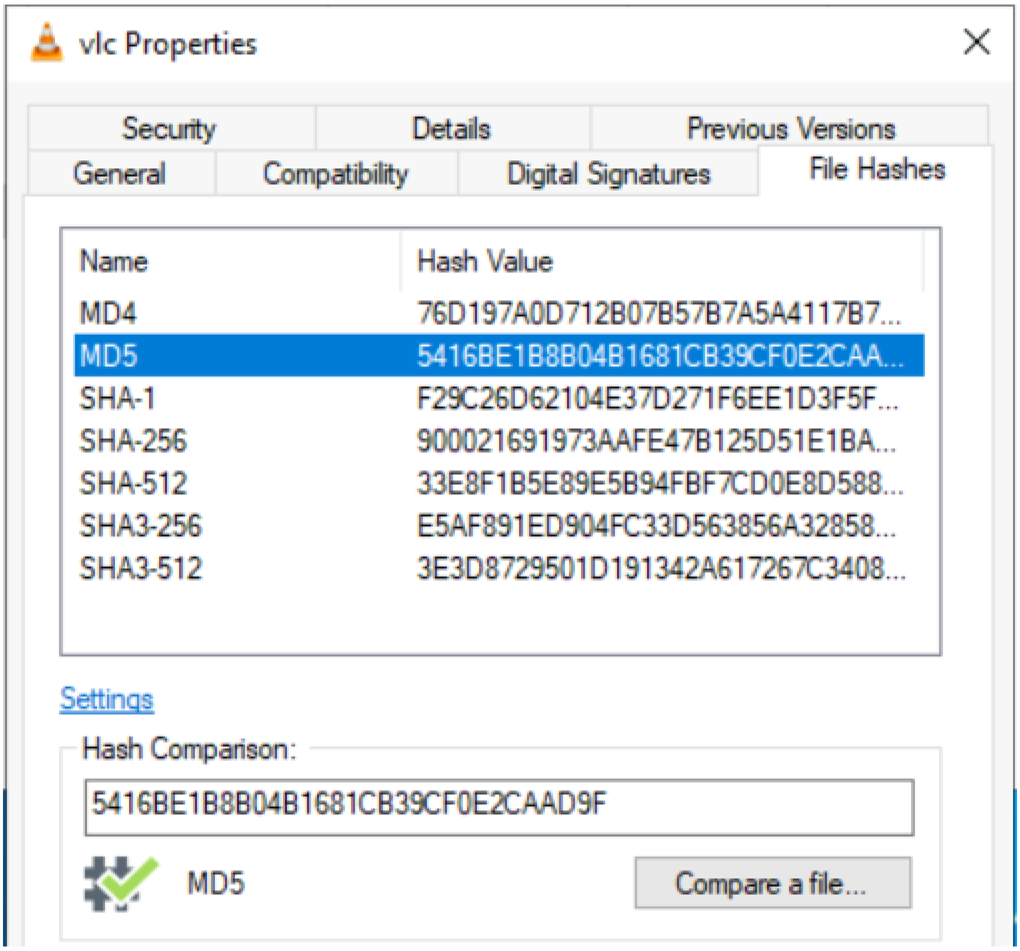
Name	Hash Value
MD4	F06C526EC3B88B594F6F8FC618BEE...
MD5	59CB421172A89E1E16C11A42832695...
SHA-1	7583DB73674F2340A3244B4E0DD9C...
SHA-256	E86EE0E2F0AEC066C3315B40F754E...
SHA-512	E79A4AFB79A6DEEEA45511A5A3749...
SHA3-256	2F163D14C88134F44C36F71C081B7D...
SHA3-512	3839C56E8AD2A703FF8514D3F6C9D...

Hash Comparison: 59CB421172A89E1E16C11A428326952C

MD5

The MD5 Checksum of vlc.exe

```
5416BE1B8B04B1681CB39CF0E2CAAD9F
```



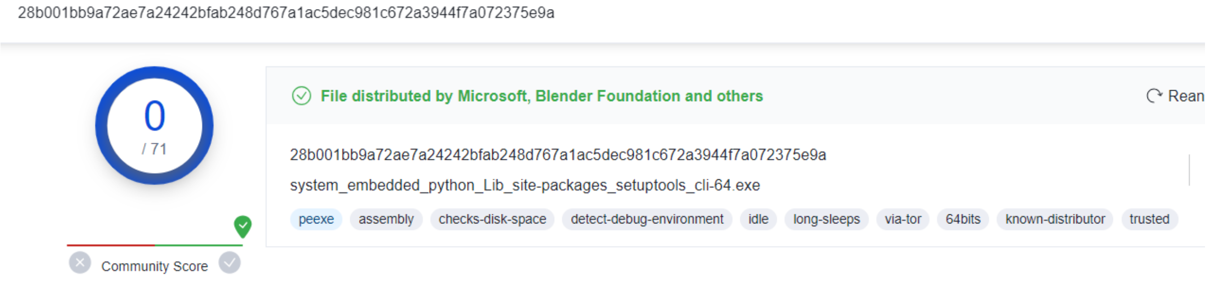
Name	Hash Value
MD4	76D197A0D712B07B57B7A5A4117B7...
MD5	5416BE1B8B04B1681CB39CF0E2CAA...
SHA-1	F29C26D62104E37D271F6EE1D3F5F...
SHA-256	900021691973AAFE47B125D51E1BA...
SHA-512	33E8F1B5E89E5B94FBF7CD0E8D588...
SHA3-256	E5AF891ED904FC33D563856A32858...
SHA3-512	3E3D8729501D191342A617267C3408...

Hash Comparison: 5416BE1B8B04B1681CB39CF0E2CAAD9F

MD5

Now lets see if the MD5 Checksums have been analysed before (Task 8)

Does Virustotal report this MD5 Checksum / file aws.exe as malicious?


28b001bb9a72ae7a24242bfab248d767a1ac5dec981c672a3944f7a072375e9a

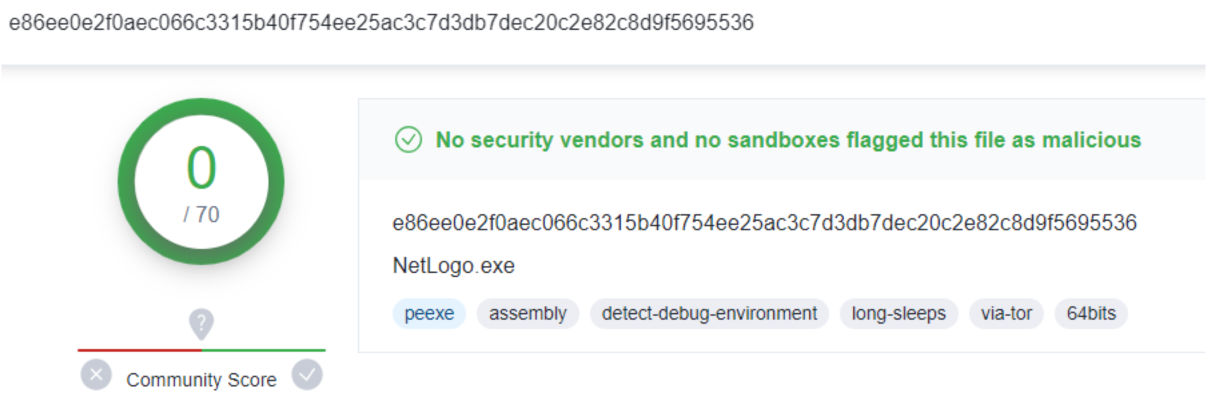
0 / 71

File distributed by Microsoft, Blender Foundation and others

28b001bb9a72ae7a24242bfab248d767a1ac5dec981c672a3944f7a072375e9a

system_embedded_python_Lib_site-packages_setuptools_cli-64.exe

Does Virustotal report this MD5 Checksum / file Netlogo.exe as malicious?


e86ee0e2f0aec066c3315b40f754ee25ac3c7d3db7dec20c2e82c8d9f5695536

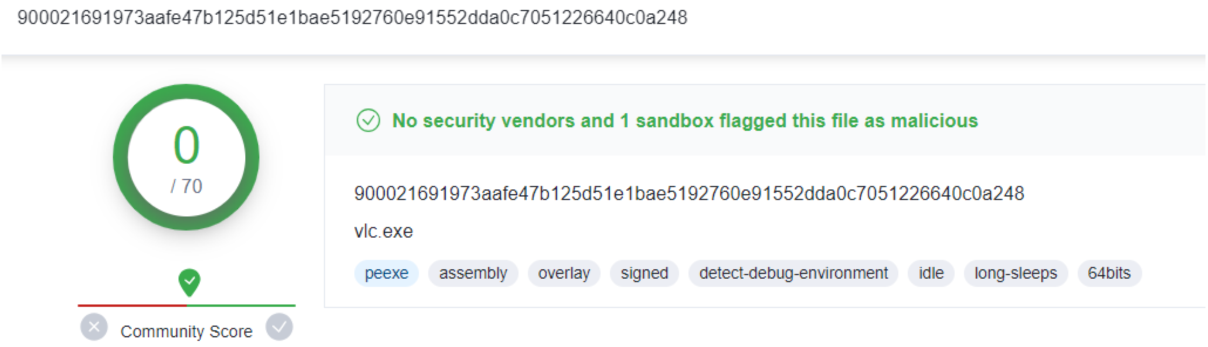
0 / 70

No security vendors and no sandboxes flagged this file as malicious

e86ee0e2f0aec066c3315b40f754ee25ac3c7d3db7dec20c2e82c8d9f5695536

NetLogo.exe

Does Virustotal report this MD5 Checksum / file vlc.exe as malicious?


900021691973aafe47b125d51e1bae5192760e91552dda0c7051226640c0a248

0 / 70

No security vendors and 1 sandbox flagged this file as malicious

900021691973aafe47b125d51e1bae5192760e91552dda0c7051226640c0a248

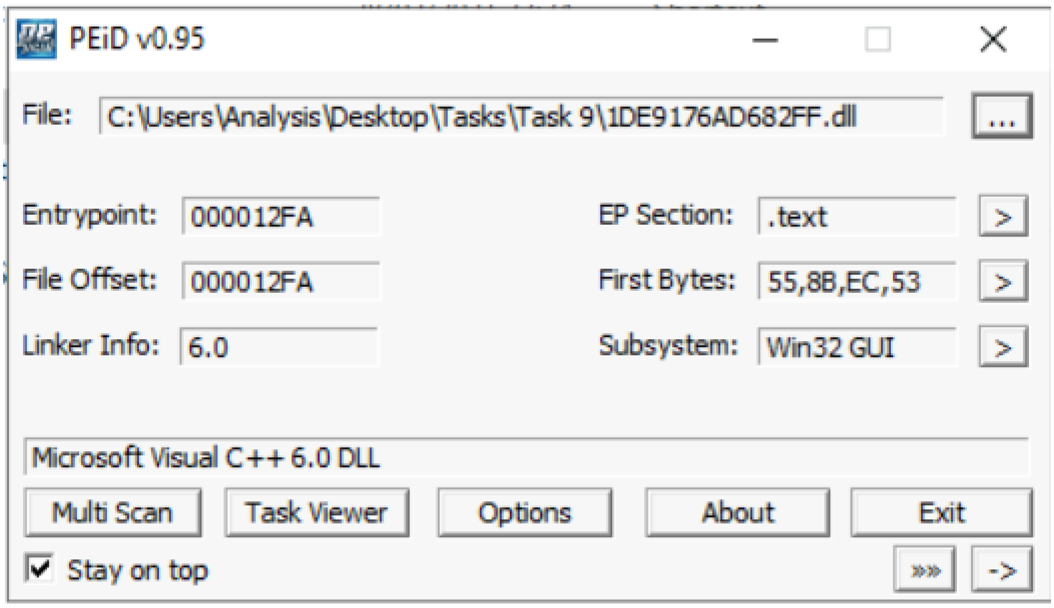
vlc.exe

Identifying if the Executables are obfuscated / packed (Task 9)

There are a few provided tools on this Windows instance that are capable of identifying the compiler / packer of a file. However, `PeID` has a huge database and is a great tool for this.
The hex value for an executable is always `4D 5A`

What does PeID propose 1DE9176AD682FF.dll being packed with?

`Microsoft Visual C++ 6.0 DLL`
Go to `Tools -> Static -> PE Tools -> PEiD`
Give the files that present in the `Task 9 in Tasks` folder.



PEiD v0.95

File: C:\Users\Analysis\Desktop\Tasks\Task 9\1DE9176AD682FF.dll

Entrypoint: 000012FA

EP Section: .text

File Offset: 000012FA

First Bytes: 55,8B,EC,53

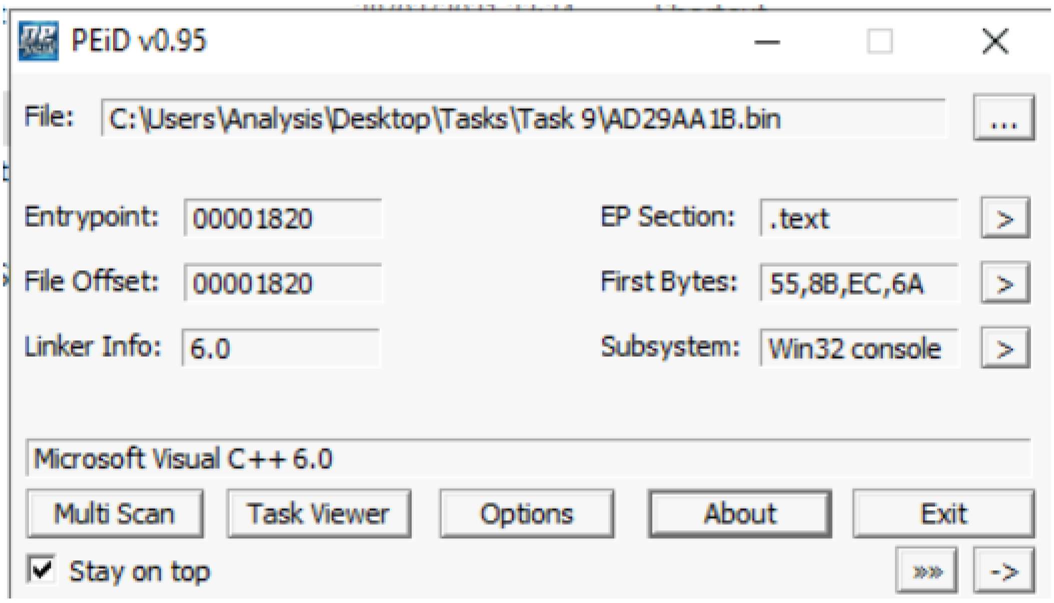
Linker Info: 6.0

Subsystem: Win32 GUI

Microsoft Visual C++ 6.0 DLL

What does PeID propose AD29AA1B.bin being packed with?

```
Microsoft Visual C++ 6.0
```



PEiD v0.95

File: C:\Users\Analysis\Desktop\Tasks\Task 9\AD29AA1B.bin

Entrypoint: 00001820

EP Section: .text

File Offset: 00001820

First Bytes: 55,8B,EC,6A

Linker Info: 6.0

Subsystem: Win32 console

Microsoft Visual C++ 6.0

What is Obfuscation / Packing? (Task 10)

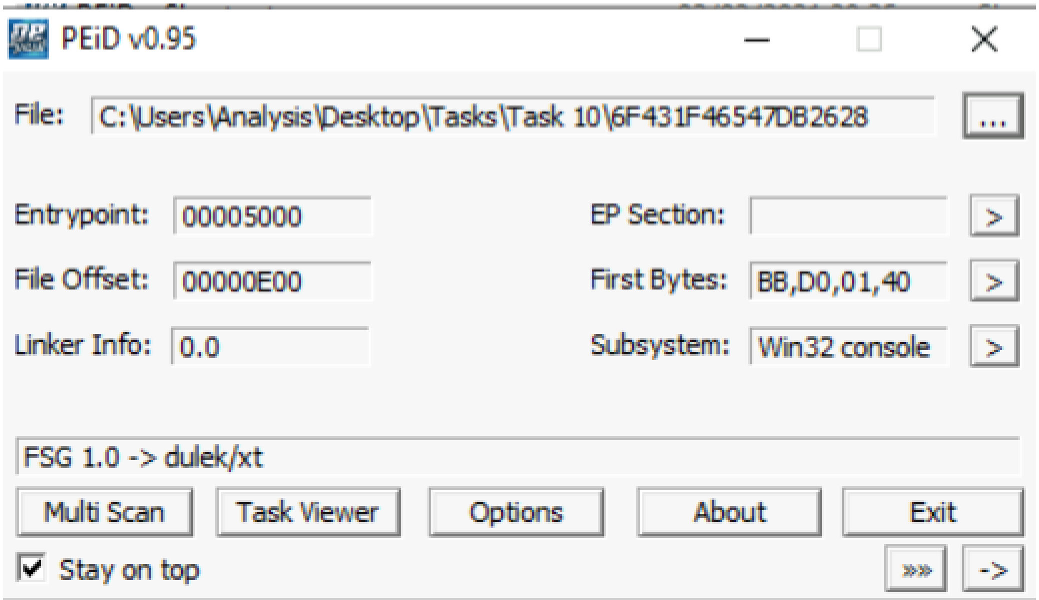
Packing is one form of obfuscation that malware Authors employ to prevent the analysis of programmes. There are both legitimate and malicious reasons as to why the Author of a program will want to prevent the decompiling of their program

Malware Authors employ obfuscation techniques such as packing - whilst for the same reasons, they do so with the intent to prevent people like us reversing it to understand its behaviours and ultimately with the aims of achieving infection.

Run the PeID to the file in `Task 10`

What packer does PeID report file "6F431F46547DB2628" to be packed with?

```
FSG 1.0 -> dulek/xt
```



File: C:\Users\Analysis\Desktop\Tasks\Task 10\6F431F46547DB2628

Entrypoint: 00005000

EP Section:

File Offset: 00000E00

First Bytes: BB,D0,01,40

Linker Info: 0.0

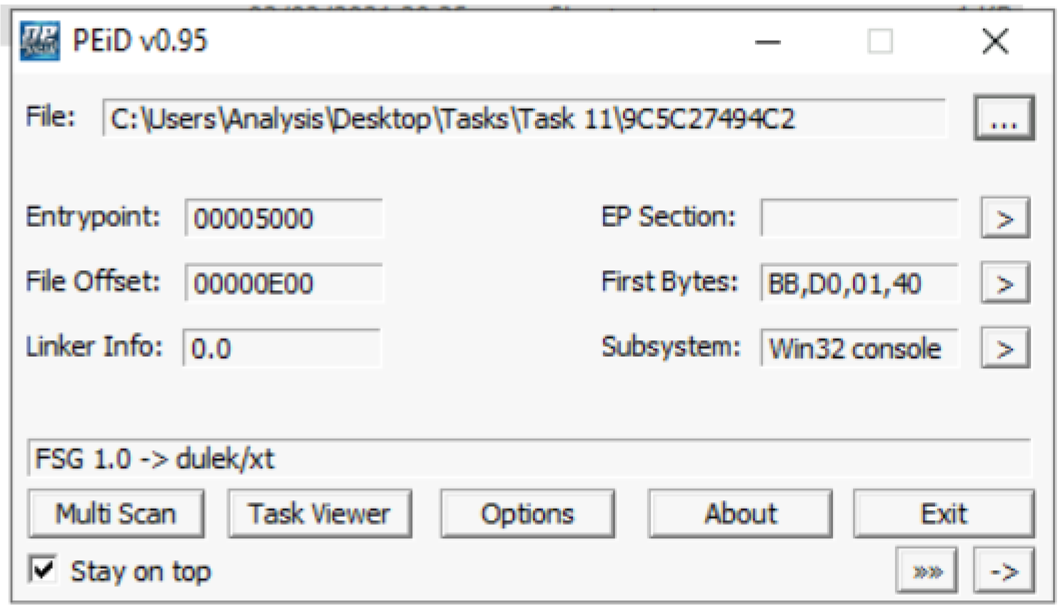
Subsystem: Win32 console

FSG 1.0 -> dulek/xt

☑ Stay on top

Visualising the Differences Between Packed & Non-Packed Code (Task 11)

Whilst PeID is capable of detecting the possibility of packers being used, it is not able to automatically de-obfuscate them. This is a process we will have to do manually - at a later stage.



File: C:\Users\Analysis\Desktop\Tasks\Task 11\9C5C27494C2

Entrypoint: 00005000

EP Section:

File Offset: 00000E00

First Bytes: BB,D0,01,40

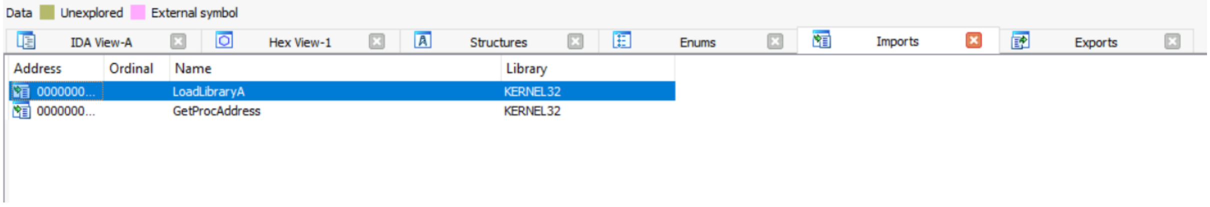
Linker Info: 0.0

Subsystem: Win32 console

FSG 1.0 -> dulek/xt

☑ Stay on top

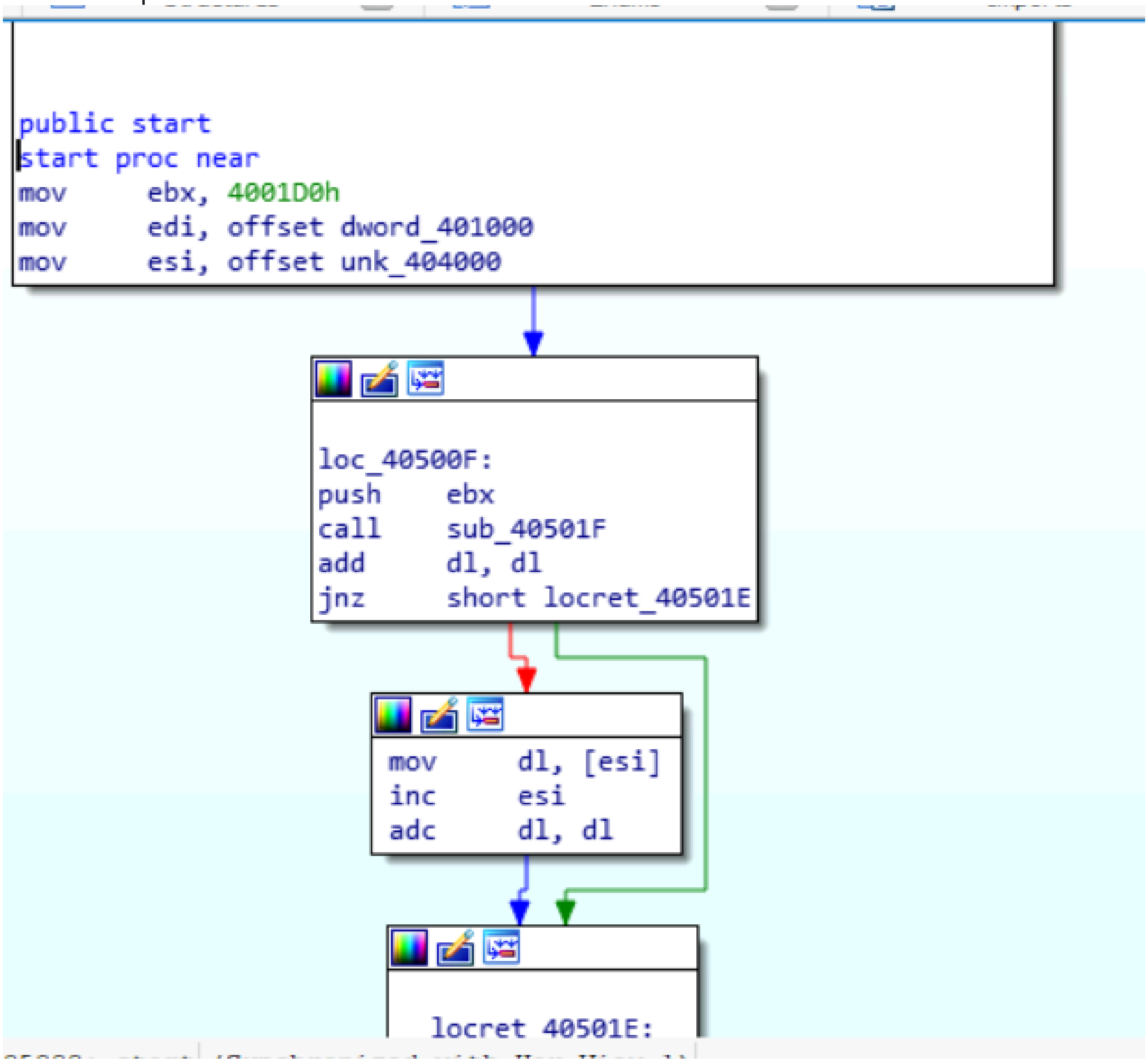
After confirming that this file is indeed packed, let's open it up with a tool called `IDA Freeware`.
We can find the tool in `Tools -> Static -> Disassembly`
Open `Imports` tab



Address	Ordinal	Name	Library
0000000...		LoadLibraryA	KERNEL32
0000000...		GetProcAddress	KERNEL32

We can see there are two Imports in this program.

We can explore futher with other tabs such as `IDA View-A`



```
public start
start proc near
mov     ebx, 4001D0h
mov     edi, offset dword_401000
mov     esi, offset unk_404000
```

```
loc_40500F:
push    ebx
call    sub_40501F
add     dl, dl
jnz     short locret_40501E
```

```
mov     dl, [esi]
inc     esi
adc     dl, dl
```

```
locret_40501E:
```

Introduction to Strings (Task 12)

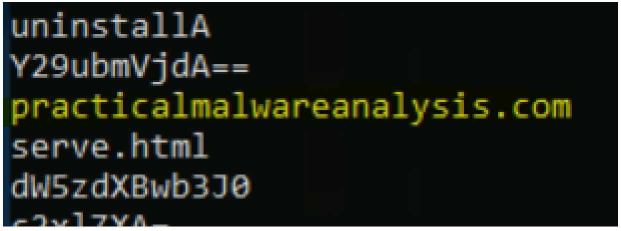
`Strings` are essentially the ASCII / Text contents of a program...this could be anything from passwords for self-extracting zips, to bitcoin addresses in ransomware samples.

Programs often contain large amount of strings and using the "strings" tool from sysinternals may only display 10% of these.

What is the URL that is outputted after using "strings"

`practicalmalwareanalysis.com`

Open `cmd` and enter `strings "C:\Users\Analysis\Desktop\Tasks\Task 12\67844C01"` in `cd C:\Users\Analysis\Desktop\Tools\SysinternalsSuite`



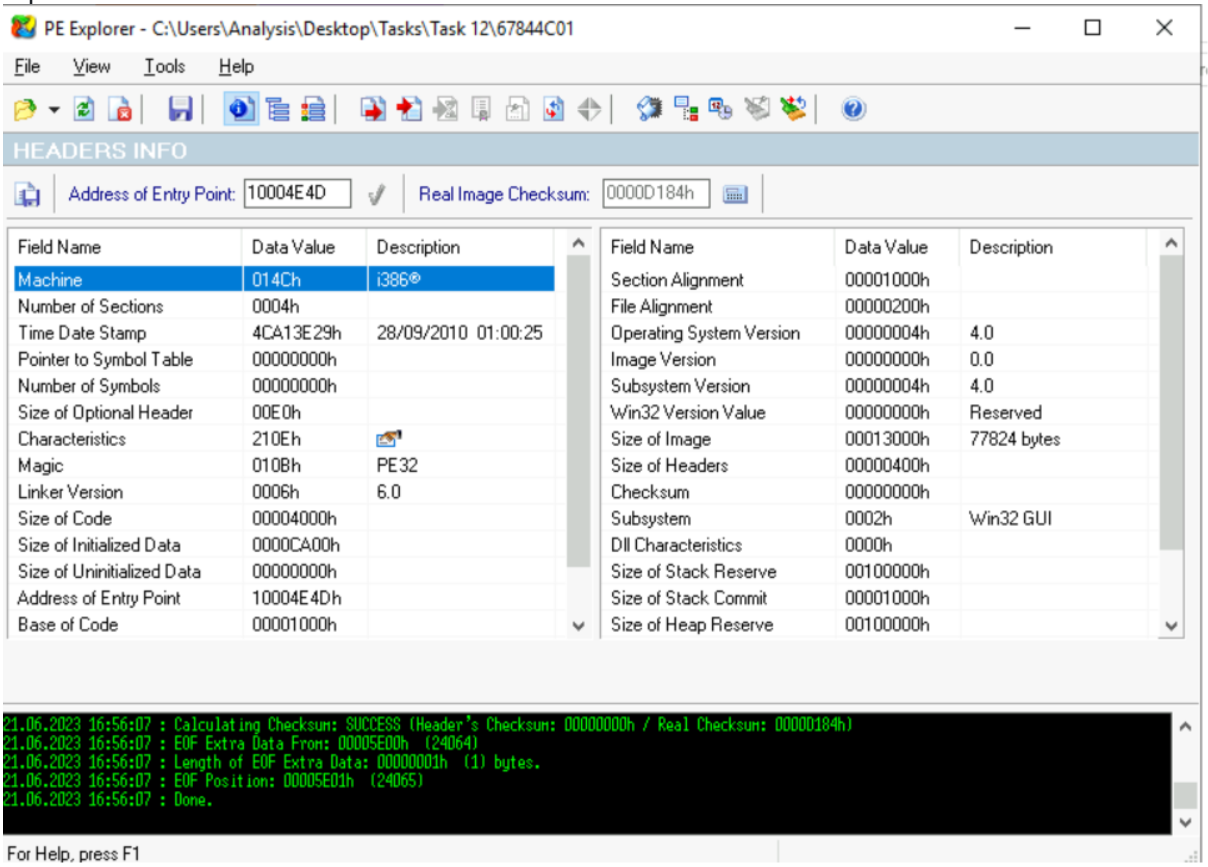
```
uninstallA
Y29ubmVjdA==
practicalmalwareanalysis.com
serve.html
dW5zdXBwb3J0
c2xlZXA=
```

How many unique "Imports" are there?

`5`

Launch the program `Tools -> Static -> PE Tools -> PE Explorer`

Open `67844C01` from the Task 12 folder



PE Explorer - C:\Users\Analysis\Desktop\Tasks\Task 12\67844C01
File
View
Tools
Help
HEADERS INFO

Address of Entry Point: 10004E4D

Real Image Checksum: 0000D184h

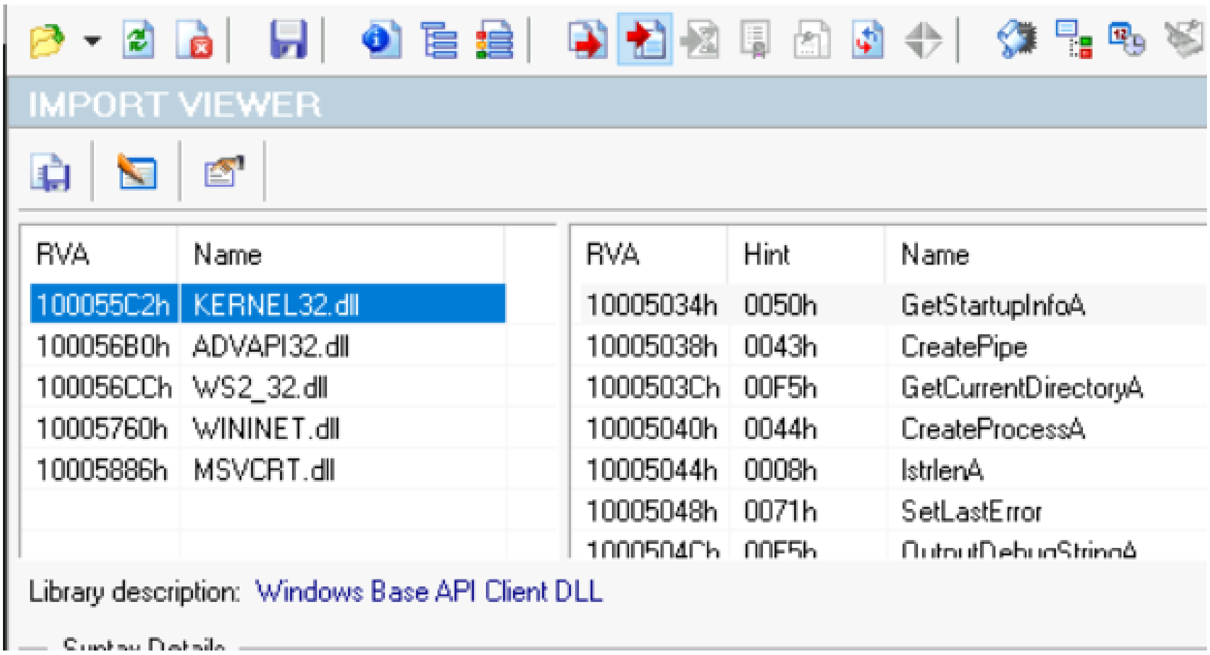
Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0004h	
Time Date Stamp	4CA13E29h	28/09/2010 01:00:25
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	210Eh	
Magic	010Bh	PE32
Linker Version	0006h	6.0
Size of Code	00004000h	
Size of Initialized Data	0000CA00h	
Size of Uninitialized Data	00000000h	
Address of Entry Point	10004E4Dh	
Base of Code	00001000h	

Field Name	Data Value	Description
Section Alignment	00001000h	
File Alignment	00000200h	
Operating System Version	00000004h	4.0
Image Version	00000000h	0.0
Subsystem Version	00000004h	4.0
Win32 Version Value	00000000h	Reserved
Size of Image	00013000h	77824 bytes
Size of Headers	00000400h	
Checksum	00000000h	
Subsystem	0002h	Win32 GUI
Dll Characteristics	0000h	
Size of Stack Reserve	00100000h	
Size of Stack Commit	00001000h	
Size of Heap Reserve	00100000h	

```
21.06.2023 16:56:07 : Calculating Checksum: SUCCESS (Header's Checksum: 00000000h / Real Checksum: 00000184h)
21.06.2023 16:56:07 : EOF Extra Data From: 00005E00h  (24064)
21.06.2023 16:56:07 : Length of EOF Extra Data: 00000001h  (1) bytes.
21.06.2023 16:56:07 : EOF Position: 00005E01h  (24065)
21.06.2023 16:56:07 : Done.
```

For Help, press F1

Go to `View` and select `Import`



RVA	Name
100055C2h	KERNEL32.dll
100056B0h	ADVAPI32.dll
100056CCh	WS2_32.dll
10005760h	WININET.dll
10005886h	MSVCRT.dll

RVA	Hint	Name
10005034h	0050h	GetStartupInfoA
10005038h	0043h	CreatePipe
1000503Ch	00F5h	GetCurrentDirectoryA
10005040h	0044h	CreateProcessA
10005044h	0008h	lstrlenA
10005048h	0071h	SetLastError
1000504Ch	00F5h	OutputDebugStringA

Library description: Windows Base API Client DLL

Syntax Details

Introduction to Imports (Task 13)

Disassemblers

Disassemblers reverse the compiled code of a program from machine code to human-readable instructions (assembly). This is limited to how the program represents itself in its current state! I.e. If the contents of an executable changes during execution - "Disassemblers" will not reflect this.

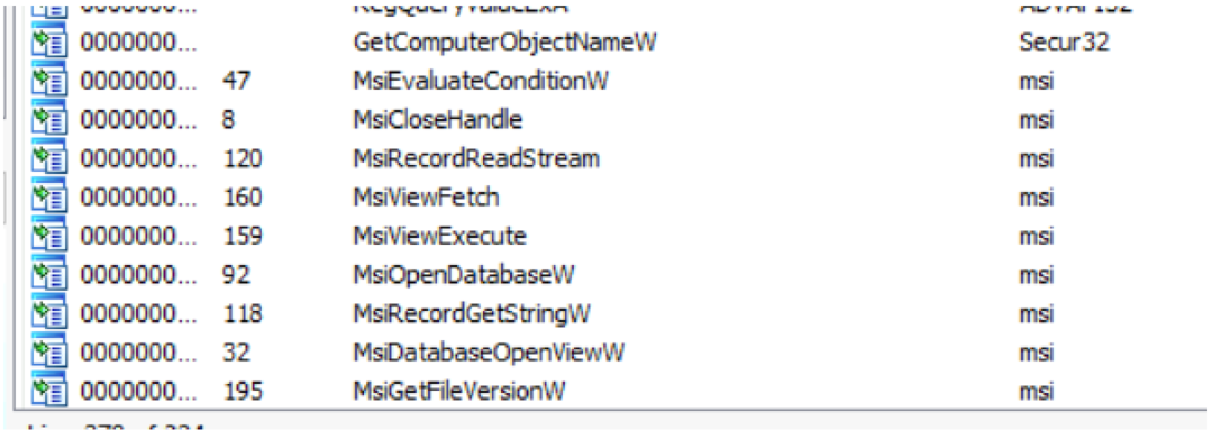
Debuggers

"Debuggers" essentially facilitate execution of the program - where the analyser can view the changes made throughout each "step" of the program. These tools are great because a true picture of the program presents itself. However, if it is indeed malicious, you have now infected yourself.

How many references are there to the library "msi" in the "Imports" tab of IDA Freeware for "install.exe"

9

Open `Tools -> Disassembly -> IDA Freeware`. Open the `install.exe` file.
Go to `Imports` tab and see the `msi` library category.



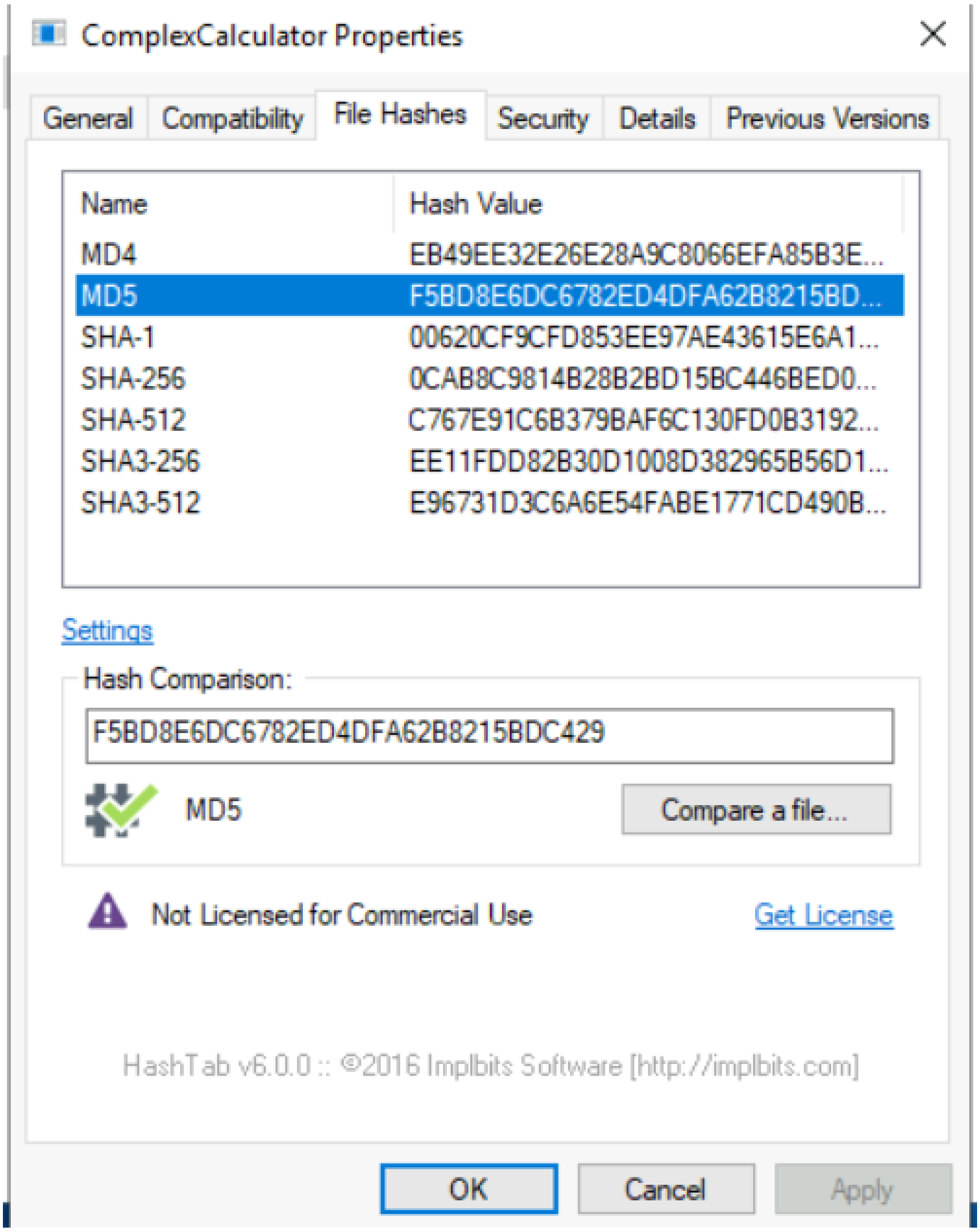
0000000...		GetComputerObjectNameW	Secur32
0000000...	47	MsiEvaluateConditionW	msi
0000000...	8	MsiCloseHandle	msi
0000000...	120	MsiRecordReadStream	msi
0000000...	160	MsiViewFetch	msi
0000000...	159	MsiViewExecute	msi
0000000...	92	MsiOpenDatabaseW	msi
0000000...	118	MsiRecordGetStringW	msi
0000000...	32	MsiDatabaseOpenViewW	msi
0000000...	195	MsiGetFileVersionW	msi

Practical Summary (Task 14)

Perform upon `ComplexCalculator.exe` in the tasks folder.

What is the MD5 Checksum of the file?

```
F5BD8E6DC6782ED4DFA62B8215BDC429
```



Name	Hash Value
MD4	EB49EE32E26E28A9C8066EFA85B3E...
MD5	F5BD8E6DC6782ED4DFA62B8215BD...
SHA-1	00620CF9CFD853EE97AE43615E6A1...
SHA-256	0CAB8C9814B28B2BD15BC446BED0...
SHA-512	C767E91C6B379BAF6C130FD0B3192...
SHA3-256	EE11FDD82B30D1008D382965B56D1...
SHA3-512	E96731D3C6A6E54FABE1771CD490B...

Hash Comparison:

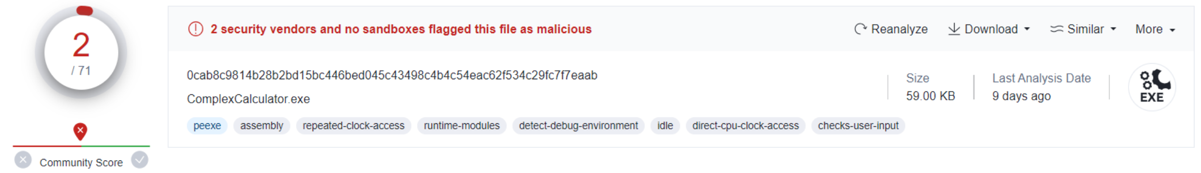
F5BD8E6DC6782ED4DFA62B8215BDC429

MD5

Not Licensed for Commercial Use

HashTab v6.0.0 :: ©2016 Implbits Software [http://implbits.com]

Does Virustotal report this file as malicious?



2 security vendors and no sandboxes flagged this file as malicious

0cab8c9814b28b2bd15bc446bed045c43498c4b4c54eac62f534c29fc7f7eaab

ComplexCalculator.exe

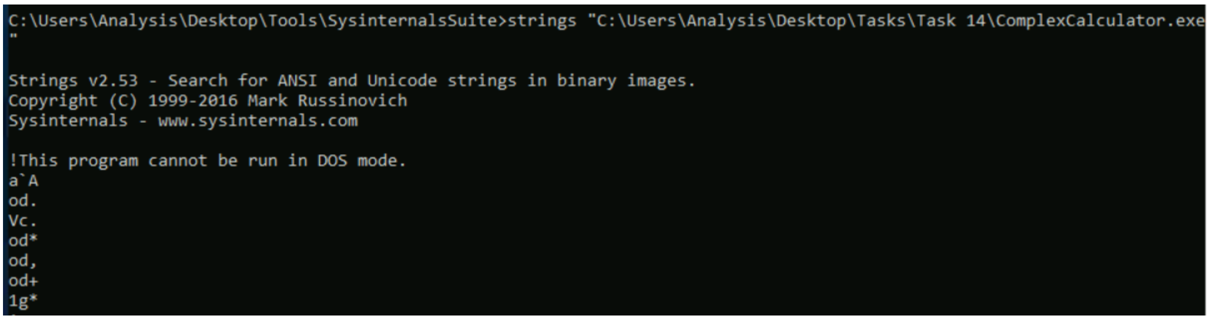
peexe assembly repeated-clock-access runtime-modules detect-debug-environment idle direct-cpu-clock-access checks-user-input

Size
59.00 KB

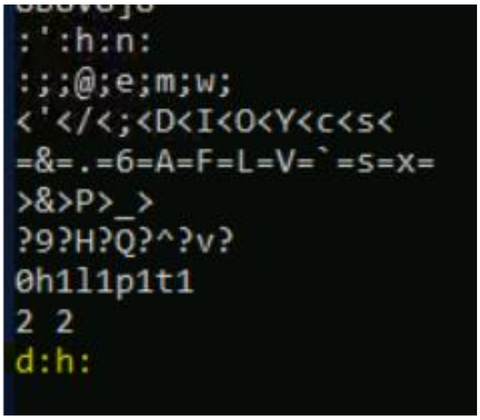
Last Analysis Date
9 days ago

Output the strings using Sysinternals "strings" tool. What is the last string outputted?

```
d:h:
```



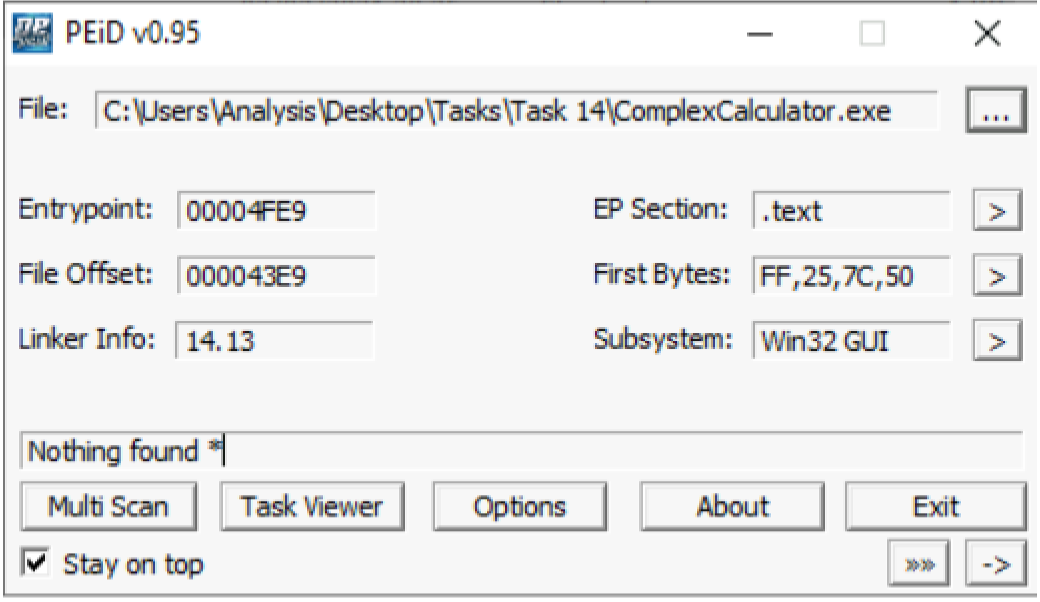
```
C:\Users\Analysis\Desktop\Tools\SysinternalsSuite>strings "C:\Users\Analysis\Desktop\Tasks\Task 14\ComplexCalculator.exe
"

Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

!This program cannot be run in DOS mode.
a`A
od.
Vc.
od*
od,
od+
1g*
```



```
:':h:n:
:;;@;e;m;w;
<'</<;<D<I<O<Y<c<s<
=&=.=6=A=F=L=V=`=s=x=
>&>P>_>
?9?H?Q?^?v?
0h1l1p1t1
2 2
d:h:
```

What is the output of PeID when trying to detect what packer is used by the file?

`Nothing Found`



PEiD v0.95

File: C:\Users\Analysis\Desktop\Tasks\Task 14\ComplexCalculator.exe

Entrypoint: 00004FE9

EP Section: .text

File Offset: 000043E9

First Bytes: FF,25,7C,50

Linker Info: 14.13

Subsystem: Win32 GUI

Nothing found *

☑ Stay on top