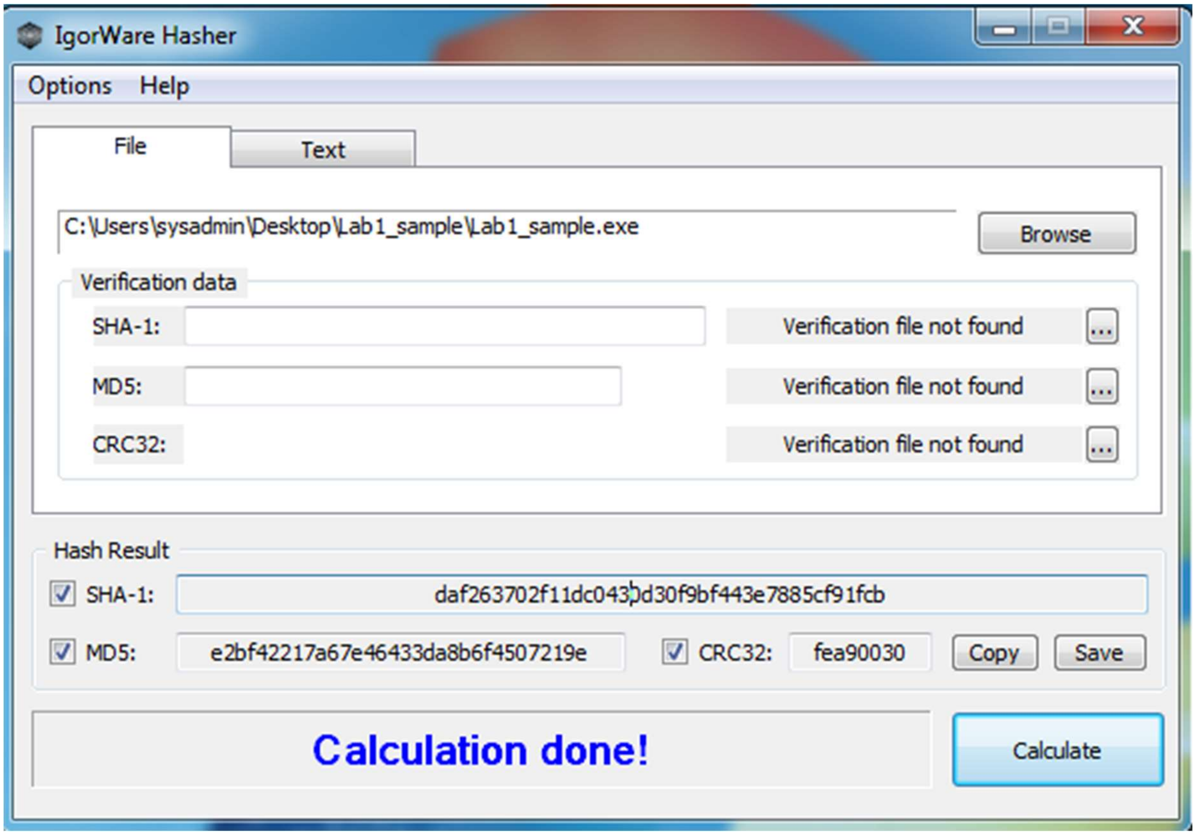
Cyber752 - Lab 1
Gunnar Yonker

**Tool:** IgoreWare Hasher

**Purpose of Tool:** This tool generates the hash value of the malware sample, this hash value can then be compared to other malware hashes and potentially find a match to a known malware and lead to identifying it. It can also be used to verify the integrity of a file by comparing the hash value of the original file with the generated hash value.

**Relevant information from tool:**



C:\Users\sysadmin\Desktop\Lab1_sample\Lab1_sample.exe

Verification data

SHA-1: Verification file not found

MD5: Verification file not found

CRC32: Verification file not found

Hash Result

☑ SHA-1: daf263702f11dc0430d30f9bf443e7885cf91fcb

☑ MD5: e2bf42217a67e46433da8b6f4507219e

☑ CRC32: fea90030

Calculation done!

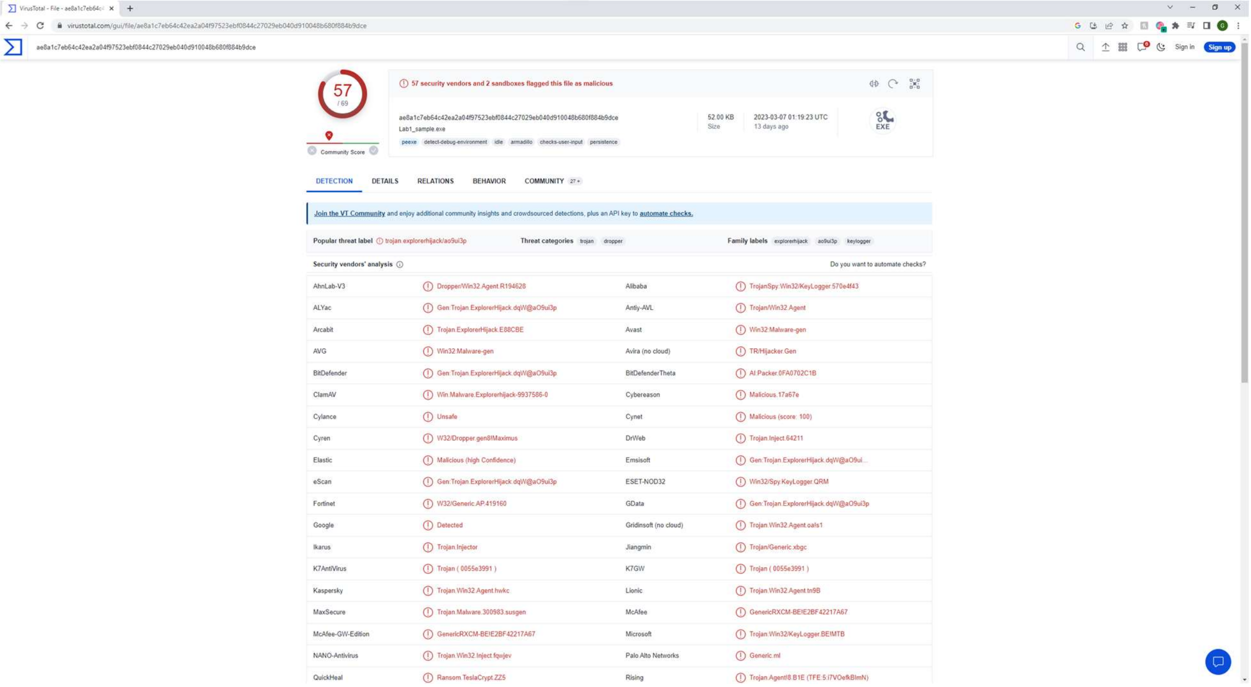
**Insights gained:** We now have the SHA-1 hash which can help to identify the malware and compare it to known malware hashes. We can also run this hash through VirusTotal to see what information can be gathered. If we suspect there are any other suspicious files on the system we can check their SHA-1 hashes and compare them to this one to see if they are the same malware that is multiplying and using different file names.

Cyber752 - Lab 1
Gunnar Yonker

**Tool:** VirusTotal

**Purpose of Tool:** This is a web-based service that analyzes malware samples using multiple different antivirus engines and tools. It then provides information on whether the sample has any detection as malware by the antivirus software, behavioral analysis, and the metadata. The user can then see these results and see if the sample is malware and potentially what kind of malware it is.

**Relevant information from tool:**


VirusTotal - File - ae8a1c7eb64c
virustotal.com/gui/file/ae8a1c7eb64c42ea2a04f97523ebf0844c27029eb040d910048b680f884b9dce
ae8a1c7eb64c42ea2a04f97523ebf0844c27029eb040d910048b680f884b9dce
Sign in
Sign up
57
/ 69
Community Score
EXE
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY 27+

57 security vendors and 2 sandboxes flagged this file as malicious

ae8a1c7eb64c42ea2a04f97523ebf0844c27029eb040d910048b680f884b9dce
Lab1_sample.exe
peexe detect-debug-environment idle armadillo checks-user-input persistence

52.00 KB
Size

2023-03-07 01:19:23 UTC
13 days ago

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

Popular threat label trojan.explorerhijack/ao9ui3p Threat categories trojan dropper Family labels explorerhijack ao9ui3p keylogger

Security vendors' analysis Do you want to automate checks?

| | | | |
|---|---|---|---|
| AhnLab-V3 | Dropper/Win32.Agent.R194628 | Alibaba | TrojanSpy:Win32/KeyLogger.570e4f43 |
| ALYac | Gen:Trojan.ExplorerHijack.dqW@aO9ui3p | Antiy-AVL | Trojan/Win32.Agent |
| Arcabit | Trojan.ExplorerHijack.E88CBE | Avast | Win32:Malware-gen |
| AVG | Win32:Malware-gen | Avira (no cloud) | TR/Hijacker.Gen |
| BitDefender | Gen:Trojan.ExplorerHijack.dqW@aO9ui3p | BitDefenderTheta | AI:Packer.0FA0702C1B |
| ClamAV | Win.Malware.Explorerhijack-9937586-0 | Cybereason | Malicious.17a67e |
| Cylance | Unsafe | Cynet | Malicious (score: 100) |
| Cyren | W32/Dropper.gen8!Maximus | DrWeb | Trojan.Inject.64211 |
| Elastic | Malicious (high Confidence) | Emsisoft | Gen:Trojan.ExplorerHijack.dqW@aO9ui... |
| eScan | Gen:Trojan.ExplorerHijack.dqW@aO9ui3p | ESET-NOD32 | Win32/Spy.KeyLogger.QRM |
| Fortinet | W32/Generic.AP.419160 | GData | Gen:Trojan.ExplorerHijack.dqW@aO9ui3p |
| Google | Detected | Gridinsoft (no cloud) | Trojan.Win32.Agent.oals1 |
| Ikarus | Trojan.Injector | Jiangmin | Trojan/Generic.xbgc |
| K7AntiVirus | Trojan ( 0055e3991 ) | K7GW | Trojan ( 0055e3991 ) |
| Kaspersky | Trojan.Win32.Agent.hwkc | Lionic | Trojan.Win32.Agent.tn9B |
| MaxSecure | Trojan.Malware.300983.susgen | McAfee | GenericRXCM-BE!E2BF42217A67 |
| McAfee-GW-Edition | GenericRXCM-BE!E2BF42217A67 | Microsoft | Trojan:Win32/KeyLogger.BE!MTB |
| NANO-Antivirus | Trojan.Win32.Inject.fqwjev | Palo Alto Networks | Generic.ml |
| QuickHeal | Ransom.TeslaCrypt.ZZ5 | Rising | Trojan.Agent!8.B1E (TFE:5:i7VOefkBlmN) |

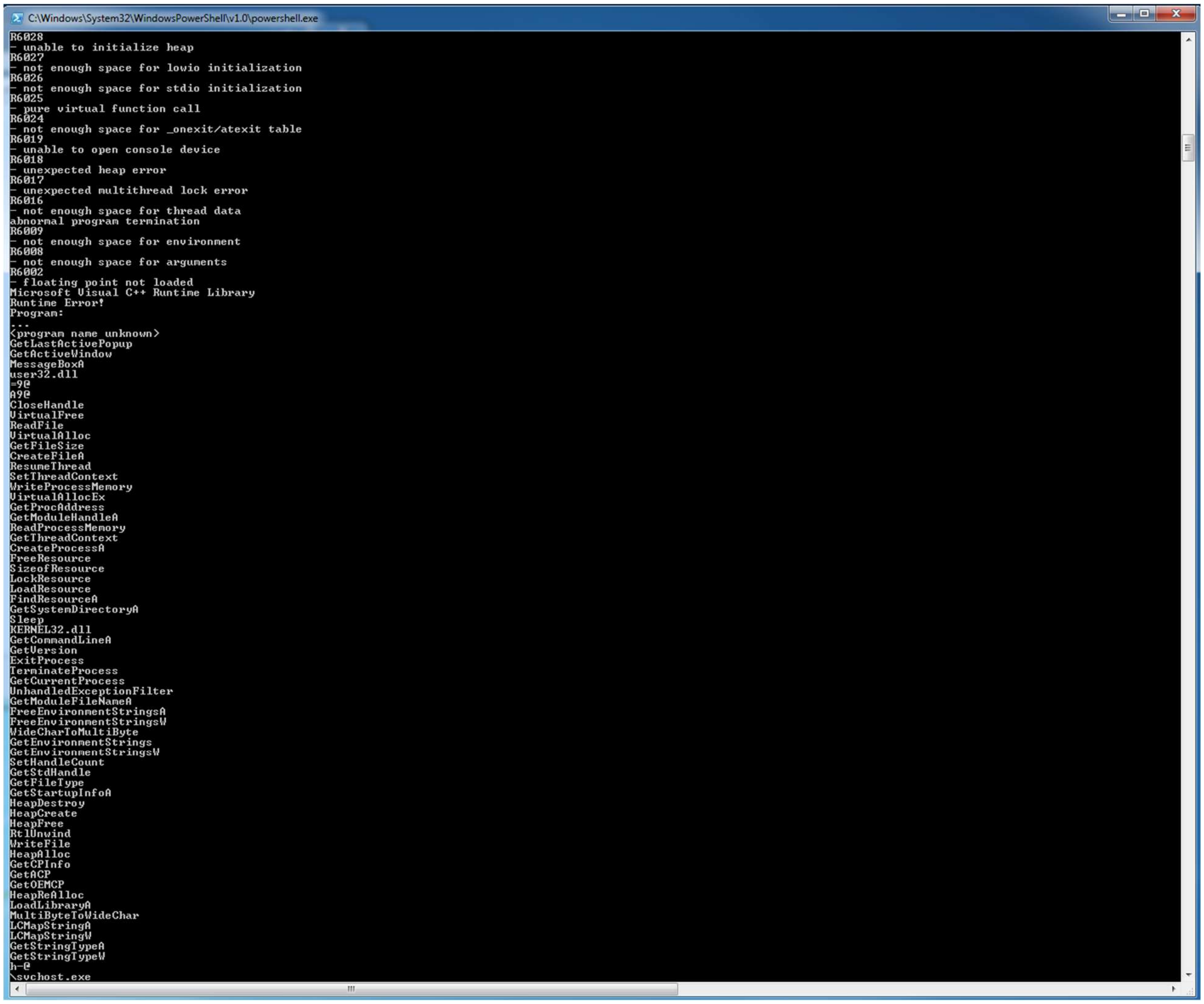
**Insights gained:** The VirusTotal scan shows that this is indeed malware as compared to many of the antivirus software programs and many of them are showing this sample as a malware specifically a trojan. A few of the reports show that it may contain a keylogger and a cryptor is also mentioned in one of the reports.

Cyber752 - Lab 1
Gunnar Yonker

**Tool:** Microsoft's strings utility

**Purpose of Tool:** This tool is used to extract the ASCII and Unicode strings from the binary files of a potential malware sample. The extracted output can include urls, registry keys, commands, and other indicators that can be used to then identify the malware.

**Relevant information from tool:** strings Lab1_sample.exe



```
R6028
- unable to initialize heap
R6027
- not enough space for lowio initialization
R6026
- not enough space for stdio initialization
R6025
- pure virtual function call
R6024
- not enough space for _onexit/atexit table
R6019
- unable to open console device
R6018
- unexpected heap error
R6017
- unexpected multithread lock error
R6016
- not enough space for thread data
abnormal program termination
R6009
- not enough space for environment
R6008
- not enough space for arguments
R6002
- floating point not loaded
Microsoft Visual C++ Runtime Library
Runtime Error!
Program:
...
<program name unknown>
GetLastActivePopup
GetActiveWindow
MessageBoxA
user32.dll
=9@
A9@
CloseHandle
VirtualFree
ReadFile
VirtualAlloc
GetFileSize
CreateFileA
ResumeThread
SetThreadContext
WriteProcessMemory
VirtualAllocEx
GetProcAddress
GetModuleHandleA
ReadProcessMemory
GetThreadContext
CreateProcessA
FreeResource
SizeofResource
LockResource
LoadResource
FindResourceA
GetSystemDirectoryA
Sleep
KERNEL32.dll
GetCommandLineA
GetVersion
ExitProcess
TerminateProcess
GetCurrentProcess
UnhandledExceptionFilter
GetModuleFileNameA
FreeEnvironmentStringsA
FreeEnvironmentStringsW
WideCharToMultiByte
GetEnvironmentStrings
GetEnvironmentStringsW
SetHandleCount
GetStdHandle
GetFileType
GetStartupInfoA
HeapDestroy
HeapCreate
HeapFree
RtlUnwind
WriteFile
HeapAlloc
GetCPInfo
GetACP
GetOEMCP
HeapReAlloc
LoadLibraryA
MultiByteToWideChar
LCMapStringA
LCMapStringW
GetStringTypeA
GetStringTypeW
h-@
\svchost.exe
```

**Insights gained:**

KERNEL32.dll – Provides an interface to the operating system, allows calls to functions

user32.dll – Same as above, both are system windows libraries being loaded

CreateFileA, ReadFile, and WriteFile - all are found as well which indicates that the malware may be trying to read or write to files on the system.

CreateProcessA, ResumeThread, SetThreadContext – Shows that the malware is trying to start new processes or manipulate existing processes.

Cyber752 - Lab 1
Gunnar Yonker

VirtualAlloc and WriteProcessMemory – Show that the malware is carrying out functions related to memory manipulation.

FreeResource, SizeofResource, LockResource, FindResourceA – The malware is trying to find resource files which can be images, files, or other data.

MultiByteToWideChar, LCMapStringA, LCMapStringW – String manipulation functions, could be trying to hide behavior.

svchost.exe reference – targeted by malware often for exploitation.

ntdll.dll – system library, may suggest that the malware is making low-level system calls.

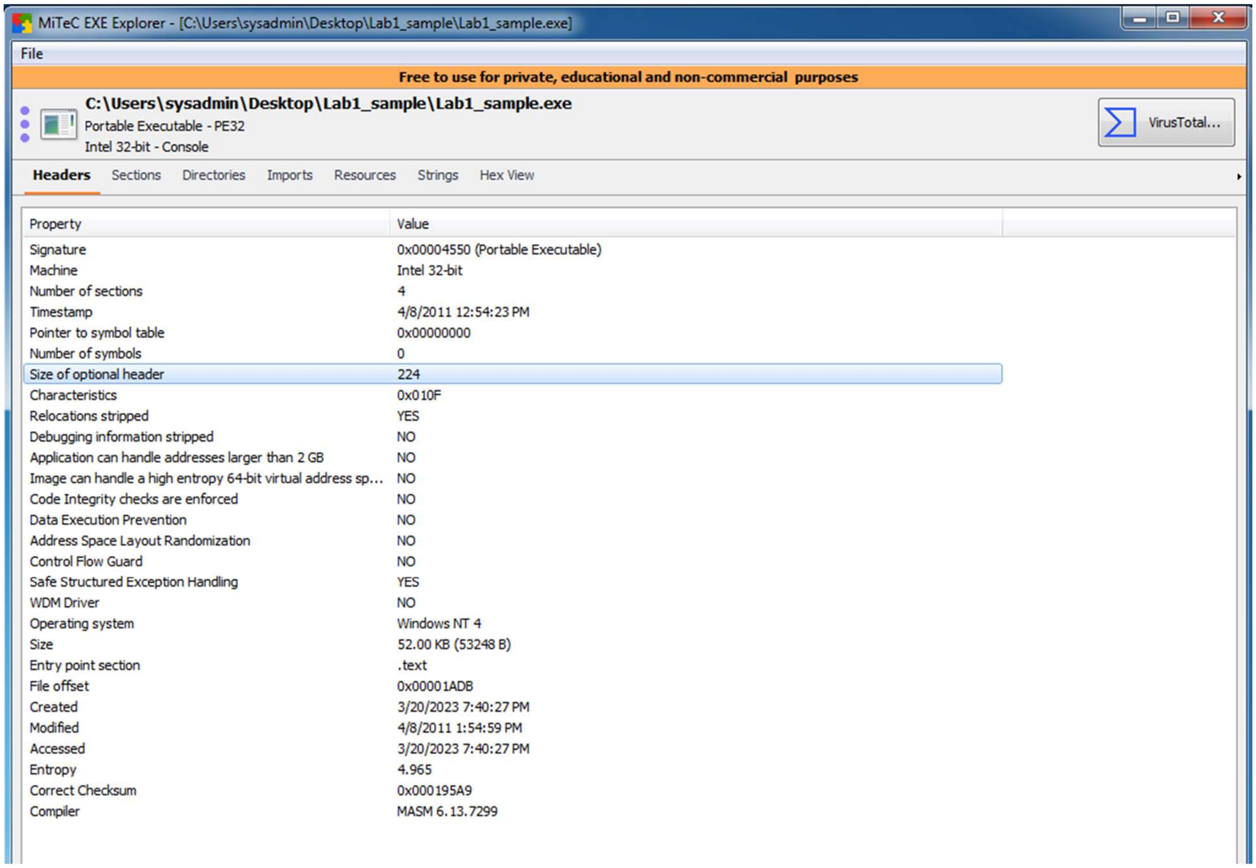
UNICODE and LOCALIZATION – this may suggest that the malware has been designed to work on a non-English version of windows possibly

Cyber752 - Lab 1
Gunnar Yonker

**Tool:** MiTec Exe Explorer

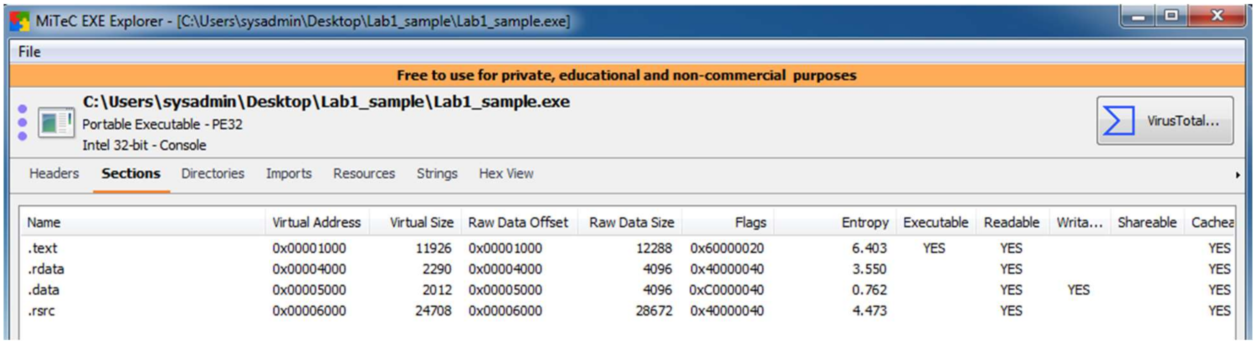
**Purpose of Tool:** This tool is a comprehensive executable file explorer that can be used to view and analyze different components of an executable file. Components such as headers, resources, imports, exports, and debug information can be gathered. This tool can also detect packers, crypters, and other forms of obfuscation.

**Relevant information from tool:**


MiTeC EXE Explorer - [C:\Users\sysadmin\Desktop\Lab1_sample\Lab1_sample.exe]
File
Free to use for private, educational and non-commercial purposes
VirusTotal...
Headers
Sections
Directories
Imports
Resources
Strings
Hex View

**C:\Users\sysadmin\Desktop\Lab1_sample\Lab1_sample.exe**
Portable Executable - PE32
Intel 32-bit - Console

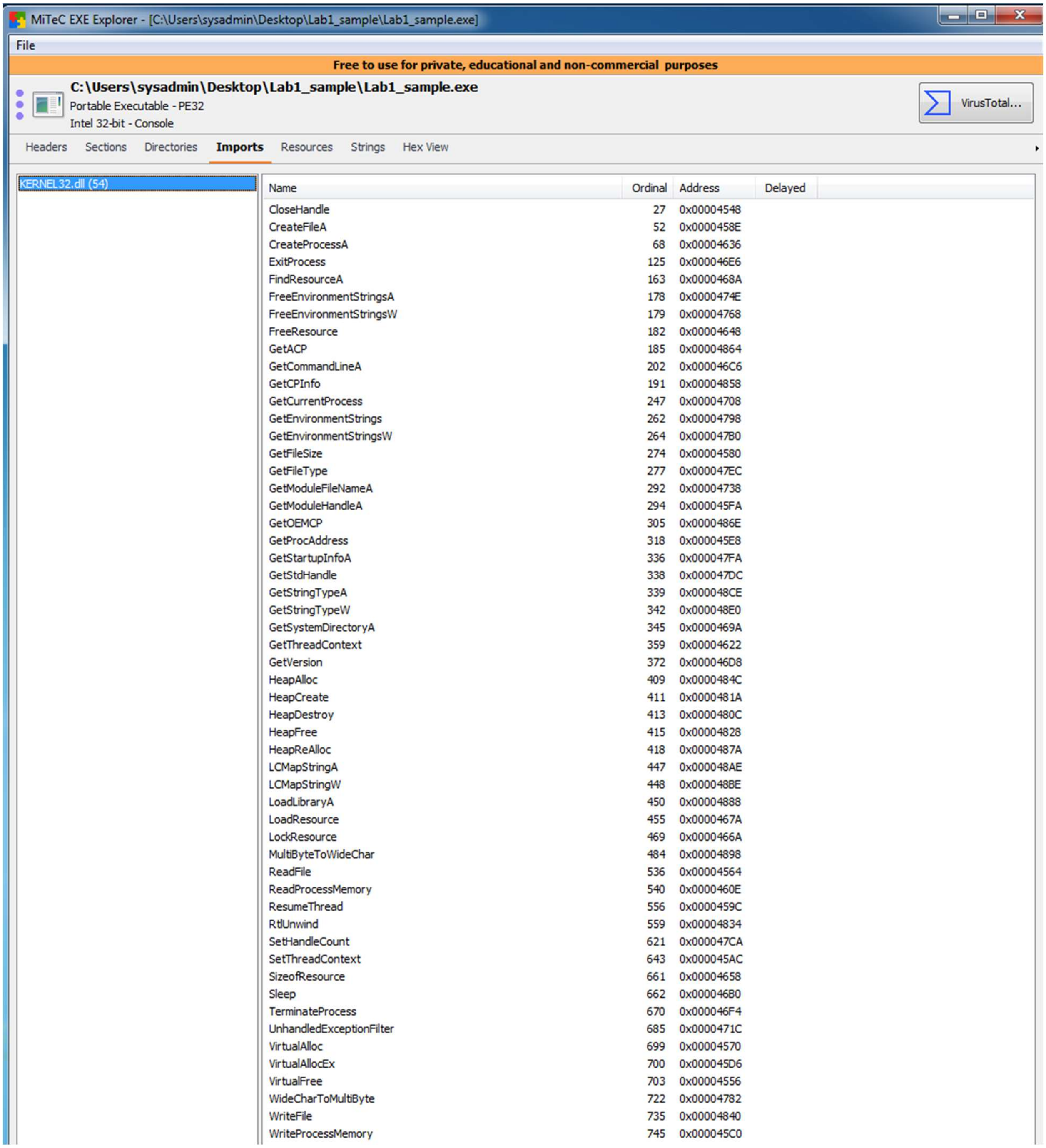
| Property | Value |
|---|---|
| Signature | 0x00004550 (Portable Executable) |
| Machine | Intel 32-bit |
| Number of sections | 4 |
| Timestamp | 4/8/2011 12:54:23 PM |
| Pointer to symbol table | 0x00000000 |
| Number of symbols | 0 |
| Size of optional header | 224 |
| Characteristics | 0x010F |
| Relocations stripped | YES |
| Debugging information stripped | NO |
| Application can handle addresses larger than 2 GB | NO |
| Image can handle a high entropy 64-bit virtual address sp... | NO |
| Code Integrity checks are enforced | NO |
| Data Execution Prevention | NO |
| Address Space Layout Randomization | NO |
| Control Flow Guard | NO |
| Safe Structured Exception Handling | YES |
| WDM Driver | NO |
| Operating system | Windows NT 4 |
| Size | 52.00 KB (53248 B) |
| Entry point section | .text |
| File offset | 0x00001ADB |
| Created | 3/20/2023 7:40:27 PM |
| Modified | 4/8/2011 1:54:59 PM |
| Accessed | 3/20/2023 7:40:27 PM |
| Entropy | 4.965 |
| Correct Checksum | 0x000195A9 |
| Compiler | MASM 6.13.7299 |


MiTeC EXE Explorer - [C:\Users\sysadmin\Desktop\Lab1_sample\Lab1_sample.exe]
File
Free to use for private, educational and non-commercial purposes
VirusTotal...
Headers
Sections
Directories
Imports
Resources
Strings
Hex View

**C:\Users\sysadmin\Desktop\Lab1_sample\Lab1_sample.exe**
Portable Executable - PE32
Intel 32-bit - Console

| Name | Virtual Address | Virtual Size | Raw Data Offset | Raw Data Size | Flags | Entropy | Executable | Readable | Writa... | Shareable | Cachea |
|---|---|---|---|---|---|---|---|---|---|---|---|
| .text | 0x00001000 | 11926 | 0x00001000 | 12288 | 0x60000020 | 6.403 | YES | YES | | | YES |
| .rdata | 0x00004000 | 2290 | 0x00004000 | 4096 | 0x40000040 | 3.550 | | YES | | | YES |
| .data | 0x00005000 | 2012 | 0x00005000 | 4096 | 0xC0000040 | 0.762 | | YES | YES | | YES |
| .rsrc | 0x00006000 | 24708 | 0x00006000 | 28672 | 0x40000040 | 4.473 | | YES | | | YES |

Cyber752 - Lab 1
Gunnar Yonker


MiTeC EXE Explorer - [C:\Users\sysadmin\Desktop\Lab1_sample\Lab1_sample.exe]
File
Free to use for private, educational and non-commercial purposes
VirusTotal...
Headers
Sections
Directories
Imports
Resources
Strings
Hex View

C:\Users\sysadmin\Desktop\Lab1_sample\Lab1_sample.exe
Portable Executable - PE32
Intel 32-bit - Console

KERNEL32.dll (54)

| Name | Ordinal | Address | Delayed |
|---|---|---|---|
| CloseHandle | 27 | 0x00004548 | |
| CreateFileA | 52 | 0x0000458E | |
| CreateProcessA | 68 | 0x00004636 | |
| ExitProcess | 125 | 0x000046E6 | |
| FindResourceA | 163 | 0x0000468A | |
| FreeEnvironmentStringsA | 178 | 0x0000474E | |
| FreeEnvironmentStringsW | 179 | 0x00004768 | |
| FreeResource | 182 | 0x00004648 | |
| GetACP | 185 | 0x00004864 | |
| GetCommandLineA | 202 | 0x000046C6 | |
| GetCPInfo | 191 | 0x00004858 | |
| GetCurrentProcess | 247 | 0x00004708 | |
| GetEnvironmentStrings | 262 | 0x00004798 | |
| GetEnvironmentStringsW | 264 | 0x000047B0 | |
| GetFileSize | 274 | 0x00004580 | |
| GetFileType | 277 | 0x000047EC | |
| GetModuleFileNameA | 292 | 0x00004738 | |
| GetModuleHandleA | 294 | 0x000045FA | |
| GetOEMCP | 305 | 0x0000486E | |
| GetProcAddress | 318 | 0x000045E8 | |
| GetStartupInfoA | 336 | 0x000047FA | |
| GetStdHandle | 338 | 0x000047DC | |
| GetStringTypeA | 339 | 0x000048CE | |
| GetStringTypeW | 342 | 0x000048E0 | |
| GetSystemDirectoryA | 345 | 0x0000469A | |
| GetThreadContext | 359 | 0x00004622 | |
| GetVersion | 372 | 0x000046D8 | |
| HeapAlloc | 409 | 0x0000484C | |
| HeapCreate | 411 | 0x0000481A | |
| HeapDestroy | 413 | 0x0000480C | |
| HeapFree | 415 | 0x00004828 | |
| HeapReAlloc | 418 | 0x0000487A | |
| LCMapStringA | 447 | 0x000048AE | |
| LCMapStringW | 448 | 0x000048BE | |
| LoadLibraryA | 450 | 0x00004888 | |
| LoadResource | 455 | 0x0000467A | |
| LockResource | 469 | 0x0000466A | |
| MultiByteToWideChar | 484 | 0x00004898 | |
| ReadFile | 536 | 0x00004564 | |
| ReadProcessMemory | 540 | 0x0000460E | |
| ResumeThread | 556 | 0x0000459C | |
| RtlUnwind | 559 | 0x00004834 | |
| SetHandleCount | 621 | 0x000047CA | |
| SetThreadContext | 643 | 0x000045AC | |
| SizeofResource | 661 | 0x00004658 | |
| Sleep | 662 | 0x000046B0 | |
| TerminateProcess | 670 | 0x000046F4 | |
| UnhandledExceptionFilter | 685 | 0x0000471C | |
| VirtualAlloc | 699 | 0x00004570 | |
| VirtualAllocEx | 700 | 0x000045D6 | |
| VirtualFree | 703 | 0x00004556 | |
| WideCharToMultiByte | 722 | 0x00004782 | |
| WriteFile | 735 | 0x00004840 | |
| WriteProcessMemory | 745 | 0x000045C0 | |

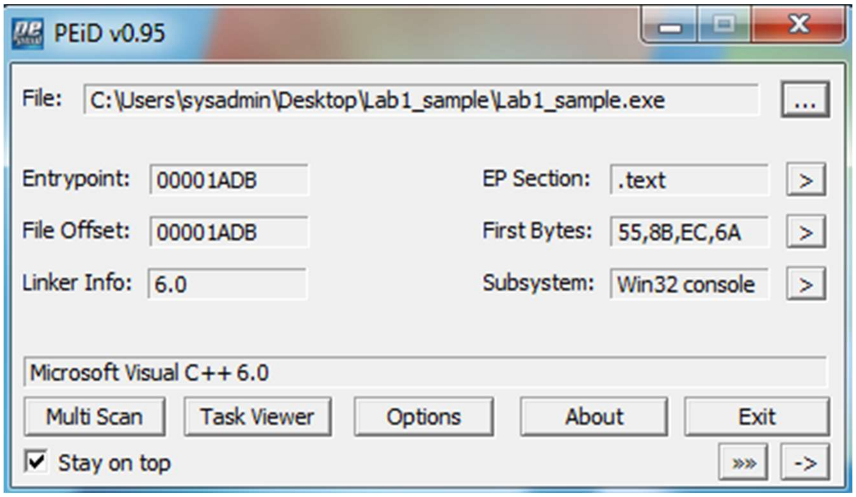
**Insights gained:** The information here shows that the timestamp is from 2011 and this indicates that this may be an older malware. There are also four sections as shown. Looking at the sections this information shows that this malware is most likely not packaged and can be confirmed using PEiD. The compiler is also shown at the end of the headers section as MASM 6.13.7299. The imports section also shows the KERNEL32.dll functions that were imported. The Strings section contains information that was previously analyzed using the powershell window and the strings command on the file.

Cyber752 - Lab 1
Gunnar Yonker

**Tool:** PEiD

**Purpose of Tool:** This tool is used to identify the packer or compiler that was used to create the malware sample by looking at the header information. This information is used to determine the approximate level of difficulty in reverse-engineering the sample and you can potentially learn what family the malware belongs to.

**Relevant information from tool:**


PEiD v0.95

File: C:\Users\sysadmin\Desktop\Lab1_sample\Lab1_sample.exe

Entrypoint: 00001ADB
EP Section: .text
File Offset: 00001ADB
First Bytes: 55,8B,EC,6A
Linker Info: 6.0
Subsystem: Win32 console

Microsoft Visual C++ 6.0

☑ Stay on top

**Insights gained:** This tool confirms that the malware sample we were given for this lab is not packed and it was compiled using Microsoft Visual C++ 6.0.

**Conclusion:**

From my analysis this malware sample was not packed and was compiled using Microsoft Visual C++ 6.0. By using IgoreWare Hasher I was able to get the SHA-1 hash which could then be searched on VirusTotal to see what the different antivirus software would detect this sample as. By using the Microsoft strings utility I was able to find a few different functions in the plaintext printout that showed the different parts of the system that would potentially be manipulated if this malware sample was executed. Further looking at the malware using MiTec EXE Explorer I was able to see that the creation date of this malware was 2011 and that it most likely was not packed given the sizes of the section. Using PEiD, it was confirmed that the malware sample was not packed as well. By using these various techniques and tools, I was able to learn that this sample is indeed malware and that it is not packed, from the strings section I learned that this malware could potentially be reading and writing file data on the system along with memory manipulation. One of the biggest insights that I had seen in the strings utility section was that there was mention of svchost.exe which is a common target of malware for exploitation.

**Takeaways:**

As for some takeaways from this lab, most of the concepts in this module are very new to me. I am familiar with the use of hashes in a different way as for the verification of integrity of a file from my Computer Forensics course that I just finished. However, now learning more uses of hash values as being able to look them up for a malware sample to see if it is a known malware, they seem even more important than they previously were in my mind. It is really interesting to me as well that there are so many different tools that can be used to analyze a malware sample that can be done without posing a large risk to the system(other than having the malware present on the system in the first place) and the different utility of each tool. I think that the string utility is a great way to learn more about the malware specifically that you can see what it might be trying to do, simply googling the functions being called to can help to provide insight into the purpose of the functions, thus what the malware might be trying to accomplish on the target system.