Malware analysis coursework 2:

# Lab11-01.exe

## Methodology

### Static analysis

1. Ran AVG antivirus on the Windows XP virtual machine. The settings were customized so that no action would be taken (for example, delete, quarantine etc.) by the antivirus software.
2. AVG antivirus found the malware and provided a file path
3. Used 7Zip on the “XP\_Victim\_for\_students” VirtualBox file to gain access to the virtual hard drive
4. Attached the extracted hard drive to the provided Lubuntu VM (Virtual Machine)
5. Booted Lubuntu virtual machine with attached XP hard drive
6. Mounted attached hard drive
7. Followed provided path on Lubuntu VM
8. Opened malware in Cutter on Lubuntu VM
9. Looked through disassembly on Cutter
10. Looked through hex dump on Cutter
11. Took MD5 hash (provided by Cutter) and searched MD5 hash in VirusTotal
12. Loaded file into IDA

### Dynamic analysis

1. Filtered ProcMon to “Lab11-01.exe” process
2. Ran WireShark
3. Took first reg shot
4. Start Process explorer
5. Started ProcMon capture
6. Started TCP view
7. Ran Lab11-01.exe
8. Took second reg shot

## Location

C:\WINDOWS\msagent\intl\MS\_PMAL\_Agent\BinaryCollection\Chapter\_11L

## Behaviour

In the VirusTotal analysis of “Lab11-01.exe” the .dll file “msgina32.dll” appeared multiple times as a file generated by “Lab11-01.exe”. I therefore decided that “msgina32.dll”. Should be first focus of my investigation

Graphical user interface, text, application, email

Description automatically generated

VirusTotal analysis showing "msgina32.dll" multiple times

Researching Gina shows that “GINA operates in the context of the Winlogon process and, as such, the GINA DLL is loaded very early in the boot process” (Microsoft, 2021). The fact that “msgina32.dll” is involved in the Winlogon process suggests that the purpose of the “Lab11-01.exe” malware is to steal user credentials. “msgina32.dll” is also likely how “Lab11-01.exe” achieves persistence.

When running “msgina32.dll” through Cutter on the Lubuntu VM, I found further information on how “Lab11-01.exe” is designed. The majority of Wlx functions shown in the screengrab below are involved in the shutdown/logoff of Windows (Microsoft, 2021). For example, “WlxIsLogoffOk” is triggered when a user triggers a log off operation and “must be implemented by a replacement GINA DLL” (Microsoft, 2021)

Graphical user interface, text, application

Description automatically generated

Reviewing the information at this point I was confident that the purpose of the malware was to steal user credentials using “msgina32.dll” and that the credentials would be stolen during the log in process

The missing part of my analysis was where credentials would be stored and how they would be transmitted to the attacker. Looking through the string analysis of cutter of “msgina32.dll”, I noticed two interesting entries – shown below.

Text

Description automatically generated

As the only string with an extension, “msutil32.sys” looked to be where the stolen login credentials would be stolen. “UN %s DM %s PW %s OLD %s” looked to be what information would be stored in “msutil32.sys” (UN for the username, DM for the domain, PW for the password and OLD unclear). Virus total showed multiple detected IP addresses on

Now that my static analysis was finished it was time to test my theories through dynamic analysis.

I loaded up ProcMon and then ran “lab11-01.exe”. I then filtered the process name to “lab11-01.exe”. As expected, “msgina32.dll” was very present in ProcMon.Table

Description automatically generated with low confidence

To test that “lab11-01.exe” would still credentials at login, I logged out of Windows XP and logged back in. As expected, the password and username were recorded in “msutil.sys” at exactly the time that I logged in – I logged off and logged back in twice to verify

A picture containing graphical user interface

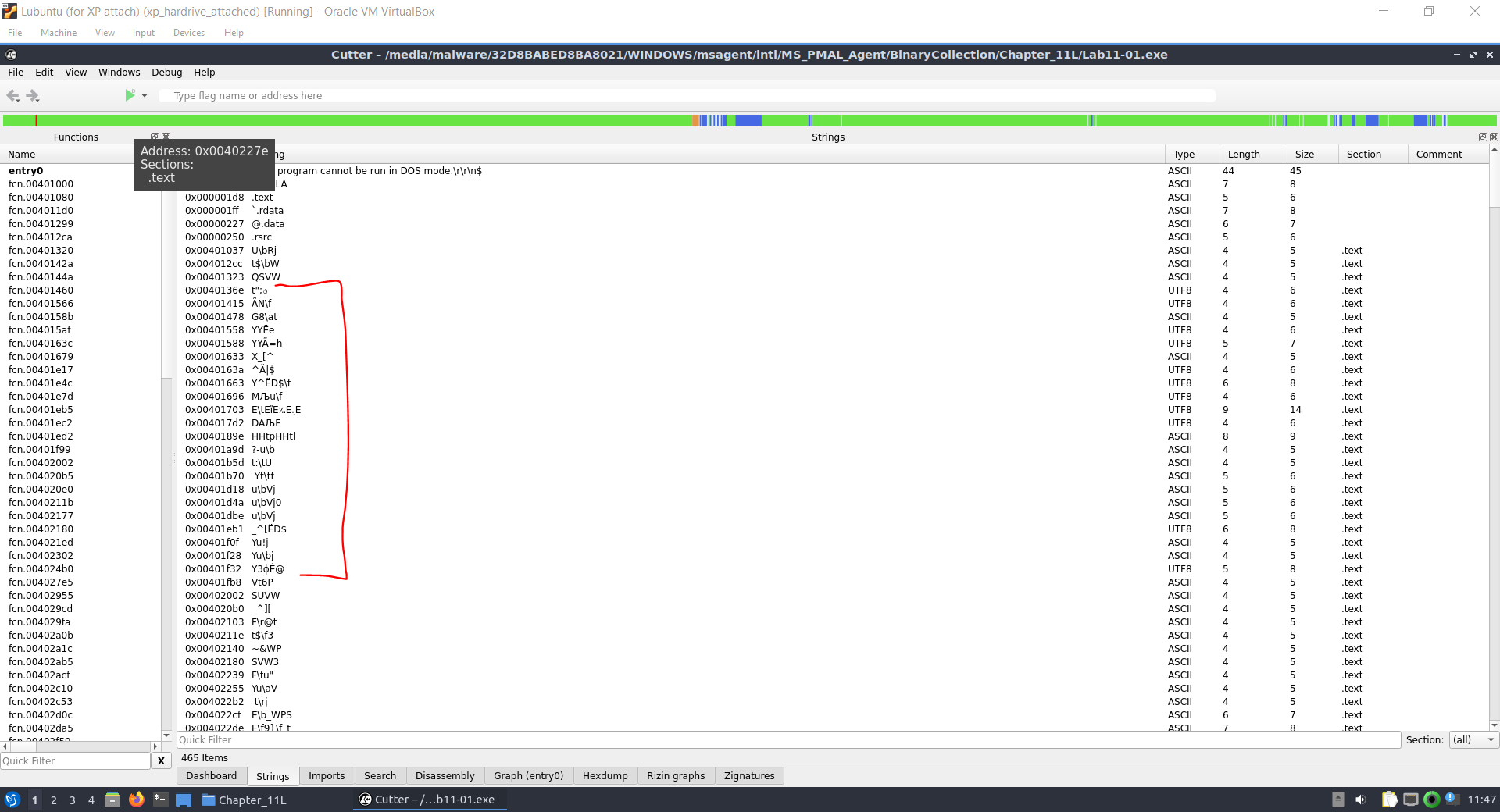
Description automatically generated

## Presence of network communication

I could not see the presence of network communication in either TCP view or WireShark

## Presence of obfuscation

Using Cutter , I looked at the strings and could clearly see the presence of obfuscation



## Suggestions for removal

Delete all files involved in the malware

# Lab11-03.exe

## Methodology

### Static analysis

1. Ran AVG antivirus on the Windows XP virtual machine. The settings were customized so that no action would be taken (for example, delete, quarantine etc.) by the antivirus software.
2. AVG antivirus found the malware and provided a file path
3. Used 7Zip on the “XP\_Victim\_for\_students” VirtualBox file to gain access to the virtual hard drive
4. Attached the extracted hard drive to the provided Lubuntu VM (Virtual Machine)
5. Booted Lubuntu virtual machine with attached XP hard drive
6. Mounted attached hard drive
7. Followed provided path on Lubuntu VM
8. Opened malware in Cutter on Lubuntu VM
9. Looked through disassembly on Cutter
10. Looked through hex dump on Cutter
11. Took MD5 hash (provided by Cutter) and searched MD5 hash in VirusTotal
12. Loaded file into IDA

### Dynamic analysis

1. Filtered ProcMon to “Lab11-03.exe” process
2. Ran WireShark
3. Took first reg shot
4. Start Process explorer
5. Started ProcMon capture
6. Started TCP view
7. Ran Lab11-03.exe
8. Took second reg shot

## Location

C:\WINDOWS\msagent\intl\MS\_PMAL\_Agent\BinaryCollection\Chapter\_11L

## Behaviour

Looking through the VirusTotal imports analysis of Lab11-03.exe. Imports such as, “HeapFree”, “HeapReAlloc”, “VirtualAlloc” and “VirtualFree” (Microsoft, 2022) (Microsoft, 2022); suggests that this malware will manipulate memory.

Loading “Lab11-03.exe” into IDA I noticed three things off interest in the main method– shown below.

Graphical user interface, text

Description automatically generated

“inet\_epar32.dll” is a .dll generated when the “Lab11-03.exe” runs the first time– rather than the .dll existing being a pre-requisite for the malware. In comparison, “lab11-03.dll” would need to be in the same file path as “Lab11-03.exe” when it is run for the malware to be successful.

“net start” is used to start a network service (Fisher, 2022). “cisvc” (Content Index Service) is a legitimate Windows XP tool that speeds up searches of files (file.net, 2022) (Kiguolis, 2021).

In the main method of IDA (before “net start cisvc” is ran), the function “sub\_401070” is ran. Using IDA to generate a graph we can see what this function is doing to “cisvc”. The functions called in “sub\_401070” are used to map “cisvc.exe” into memory in order to read and write to it (Microsoft, 2022) (Microsoft, 2022) (Microsoft, 2022)

Diagram

Description automatically generated

The next focus of my analysis would be “Lab11-03.dll”.

Using VirusTotal, I took a look at the imports for “lab11-03.dll”. One of the imports is GetAsyncKeyState which is used to detect “whether a key is up or down at the time the function is called” (Microsoft, 2022). In malware, GetAsyncKeyState is typically used for key loggers (Honig & Sikorski, 2012)

Looking through the cutter analysis of “Lab11-03.dll”, I noticed the string “C:\\WINDOWS\\System32\\kernel64x.dll”. This stood out to me as a file path was present suggesting that “kernel64x.dll” would be created during the runtime of “Lab11-03.dll” instead of being a pre-requisite. I will explore this further in my dynamic analysis

I ran “lab11-03.exe”. Using a tool called search everything that allows rapid searching of files I looked up “kernel64x.dll” and noticed something interesting. Compared to the other .dll files in the same directory “kernel64x.dll” had a very small file size. Opening up “kernel64x.dll” with notepad revealed that this was where the keylogging information was contained

Text

Description automatically generated

“kernel64x.dll” compared to other .dll files

Text

Description automatically generated

3The contents of “kernel64x.dll”

## Presence of network communication

I could not see the presence of network communication in either TCP view or WireShark

## Presence of obfuscation

Using Cutter , I looked at the strings and could clearly see the presence of obfuscation

Graphical user interface, text, application

Description automatically generated

## Suggestions for removal

Delete all files involved in the malware

# Lab12-03.exe

## Methodology

### Static analysis

1. Ran AVG antivirus on the Windows XP virtual machine. The settings were customized so that no action would be taken (for example, delete, quarantine etc.) by the antivirus software.
2. AVG antivirus found the malware and provided a file path
3. Used 7Zip on the “XP\_Victim\_for\_students” VirtualBox file to gain access to the virtual hard drive
4. Attached the extracted hard drive to the provided Lubuntu VM (Virtual Machine)
5. Booted Lubuntu virtual machine with attached XP hard drive
6. Mounted attached hard drive
7. Followed provided path on Lubuntu VM
8. Opened malware in Cutter on Lubuntu VM
9. Looked through disassembly on Cutter
10. Looked through hex dump on Cutter
11. Took MD5 hash (provided by Cutter) and searched MD5 hash in VirusTotal
12. Loaded file into IDA

### Dynamic analysis

1. Filtered ProcMon to “Lab12-03.exe” process
2. Ran WireShark
3. Took first reg shot
4. Start Process explorer
5. Started ProcMon capture
6. Started TCP view
7. Ran Lab12-03.exe
8. Took second reg shot

## Location

C:\WINDOWS\msagent\intl\MS\_PMAL\_Agent\BinaryCollection\Chapter\_12L

## Behaviour

Running “Lab12-03.exe” through VirusTotal I took a look at the imports. One import that stood out to me was SetWindowsHookExA which allows applications to define hooks that can monitor the system for certain types of events (Microsoft, 2022)

Loading “Lab12-03.exe” and taking a look at the main function. I noticed the code shown below

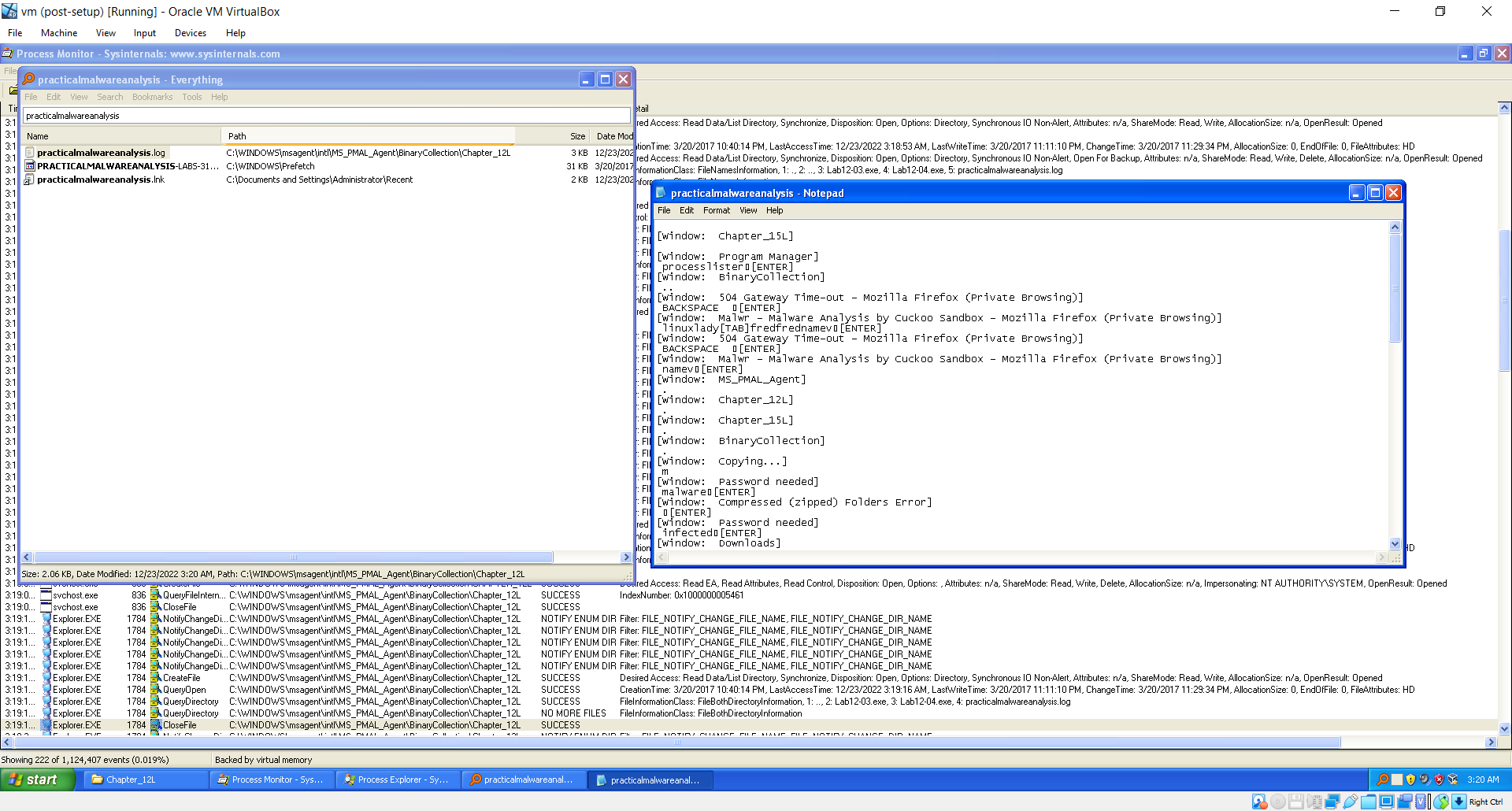
Text

Description automatically generated with low confidence

The line “0Dh” is used to specify which hook should be used by SetWindowsHookExA. When converted from hex to decimal, “0D” is equal to 13. 13 refers to the WH\_KEYBOARD\_LL hook (Microsoft, 2022). This indicates that this malware is a keylogger.

Looking through the string section of Cutter I found “practicalmalwareanalysis.log”. As the only string with an extension, this is likely where keylogging information is stored.

I ran “Lab12-03.exe” on the XP virtual machine and used a tool called search everything to find “practicalmalwareanalysis.log”. As expected “practicalmalwareanalysis.log” stores keylogging information.



## Presence of network communication

I could not see the presence of network communication in either TCP view or WireShark

## Presence of obfuscation

Using Cutter , I looked at the strings and could clearly see the presence of obfuscation

Graphical user interface, text, application, email

Description automatically generated

## Suggestions for removal

Delete all files involved in the malware

# Lab12-04.exe

## Methodology

### Static analysis

1. Ran AVG antivirus on the Windows XP virtual machine. The settings were customized so that no action would be taken (for example, delete, quarantine etc.) by the antivirus software.
2. AVG antivirus found the malware and provided a file path
3. Used 7Zip on the “XP\_Victim\_for\_students” VirtualBox file to gain access to the virtual hard drive
4. Attached the extracted hard drive to the provided Lubuntu VM (Virtual Machine)
5. Booted Lubuntu virtual machine with attached XP hard drive
6. Mounted attached hard drive
7. Followed provided path on Lubuntu VM
8. Opened malware in Cutter on Lubuntu VM
9. Looked through disassembly on Cutter
10. Looked through hex dump on Cutter
11. Took MD5 hash (provided by Cutter) and searched MD5 hash in VirusTotal
12. Loaded file into IDA

### Dynamic analysis

1. Filtered ProcMon to “Lab12-04.exe” process
2. Ran WireShark
3. Took first reg shot
4. Start Process explorer
5. Started ProcMon capture
6. Started TCP view
7. Ran Lab12-04.exe
8. Took second reg shot

## Location

C:\WINDOWS\msagent\intl\MS\_PMAL\_Agent\BinaryCollection\Chapter\_12L

## Behaviour

When “lab12-04.exe” was ran through VirusTotal, the import CreateRemoteThread stood out to me. CreateRemoteThread is used to “Creates a thread that runs in the virtual address space of another process.” (Microsoft, 2022). This gives a clue that “lab12-04.exe” likely hijacks a legitimate process.

Two other imports: LoadResource and FindResourceA are used for resource manipulation (Sikorski & Honig, 2012) (Microsoft, 2022)

Looking through the string section of Cutter on the Lubuntu VM there are some interesting entries.

Text

Description automatically generated with medium confidence

“winlogin.exe” is a legitimate Windows process that is involved in the Windows sign in process and is usually contained in the “System32” directory. However, if “winlogin” is present in a subfolder of system32 (or any other directory) it can be considered malicious

“winlogon.exe” is a legitimate Windows process used for signing in and out of Windows (Hoffman, 2017)

“winup.exe” on the internet it is considered a malicious program and is “able to monitor applications, connect to the Internet, record keyboard and mouse inputs and manipulate other programs” (BLEEPINGCOMPUTER, 2022) (file.net, 2022).

”wupdmgr.exe” is another legitimate Windows process that handles updates. However, according to file.net, if ”wupdmgr.exe” is located in a subfolder of “C:\Windows” it can be considered dangerous (file.net, 2022) and is utilised in keyloggers (BLEEPINGCOMPUTER, 2022). As seen in the string above “wupdmgr.exe” is entered into “system32” (a subfolder of “C:\Windows”). ”wupdmgr.exe” has the ability to “listens for or sends data on open ports to a LAN or the Internet” (file.net, 2022)

We can draw some conclusions from the information we have gathered:

* The disassembly of “lab12-04.exe” referencing multiple legitimate Windows processes and having the import CreateRemoteThread means that “lab12-04.exe” maliciously utilizes these legitimate processes
* “lab12-04.exe” uses “winup.exe” to manipulate the legitimate Window processes
* ”wupdmgr.exe” is likely what “lab12-04.exe” uses to connect to the internet

Another interesting string is the web address “http://www.practicalmalwareanalysis.com/updater.exe”. From the .exe extension it seems that the purpose of connecting to this URL is to download “updater.exe” – and likely the purpose of “lab12-04.exe”.

What is still unclear to me is what “winlogon.exe” is used for.

Opened up ProcMon and then ran “lab12-04.exe” and then filtered ProcMon to the process name of winup.exe. As shown in the image below, “lab12-04.exe” creates the file “winup.exe” in the TEMP directory

Graphical user interface, text, application, email

Description automatically generated

Filtering ProcMon to “wupdmgr.exe” we can see that “winup.exe” starts to manipulate “wupdmgr.exe”.

Graphical user interface, text, application

Description automatically generated

Looking through the “wupdmgr.exe” processes I can also see “wupdmgr.exe” accessing [www.practicalmalwareanalysis.com](http://www.practicalmalwareanalysis.com) to download the file

A picture containing graphical user interface

Description automatically generated

To find the IP address of [www.practicalmalwareanalysis.com](http://www.practicalmalwareanalysis.com), I used the ping utility in windows command prompt

Text

Description automatically generated

I then filtered the IP address in WireShark and could clearly see “updater.exe” being downloaded from the address

Graphical user interface, text, application, email

Description automatically generated

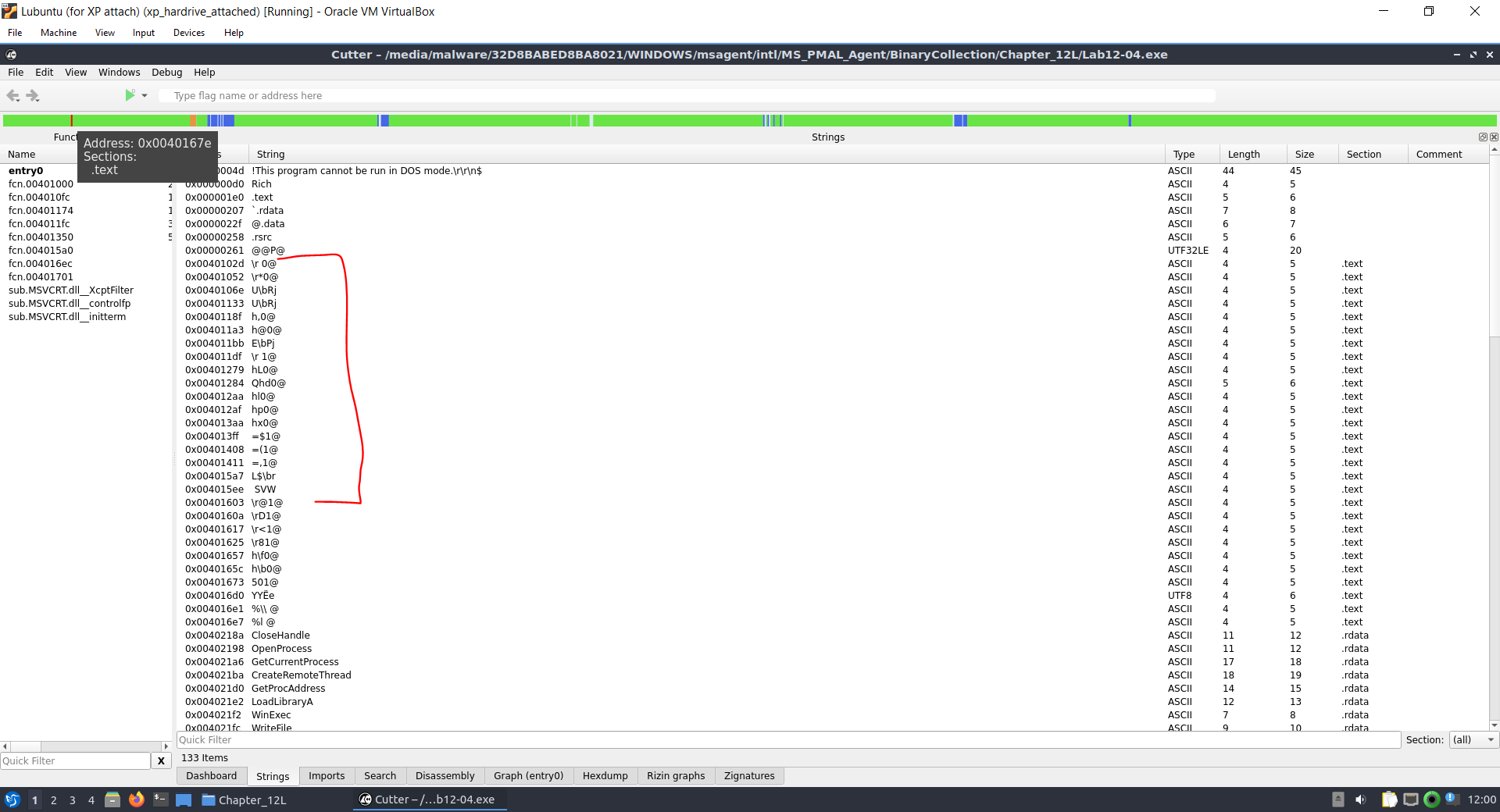
## Presence of network communication

<http://www.practicalmalwareanalysis.com/> is used in the malware. I used the ping utility on this address and filtered it in WireShark. Shown below is all the network communication used by the malware. In particular, the malware downloading a file from the internetGraphical user interface, text, application, email

Description automatically generated

## Presence of obfuscation

Using Cutter , I looked at the strings and could clearly see the presence of obfuscation



## Suggestions for removal

Delete all files involved in the malware

# Bibliography

BLEEPINGCOMPUTER, 2022. *WINUP.EXE Information.* [Online]   
Available at: https://www.bleepingcomputer.com/startups/winup.exe-18972.html  
[Accessed 19 12 2022].

BLEEPINGCOMPUTER, 2022. *WUPDMGR.EXE Information.* [Online]   
Available at: https://www.bleepingcomputer.com/startups/wupdmgr.exe-12357.html  
[Accessed 18 12 2022].

file.net, 2022. *What is cisvc.exe?.* [Online]   
Available at: https://www.file.net/process/cisvc.exe.html  
[Accessed 04 12 2022].

file.net, 2022. *Winup.exe file information.* [Online]   
Available at: https://www.file.net/process/winup.exe.html  
[Accessed 19 12 2022].

file.net, 2022. *Wupdmgr.exe file information.* [Online]   
Available at: https://www.file.net/process/wupdmgr.exe.html  
[Accessed 19 12 2022].

Fisher, T., 2022. *Net Command.* [Online]   
Available at: https://www.lifewire.com/net-command-2618094  
[Accessed 04 12 2022].

Hoffman, C., 2017. *What Is Windows Logon Application (winlogon.exe), and Why Is It Running on My PC?.* [Online]   
Available at: https://www.howtogeek.com/322411/what-is-windows-logon-application-winlogon.exe-and-why-is-it-running-on-my-pc/  
[Accessed 20 12 2022].

Honig, A. & Sikorski, M., 2012. User-Space Keyloggers. In: *Practical malware analysis : the hands-on guide to dissecting malicious software.* San Francisco: No Starch Press, p. 239.

Kiguolis, U., 2021. *What is cisvc.exe? Should I remove it?.* [Online]   
Available at: https://www.2-spyware.com/file-cisvc-exe.html  
[Accessed 04 12 2022].

Microsoft, 2021. *GINA.* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/secauthn/gina  
[Accessed 03 12 2022].

Microsoft, 2021. *winwlx.h header.* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/winwlx/  
[Accessed 03 12 2022].

Microsoft, 2021. *WlxIsLogoffOk function (winwlx.h).* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/winwlx/nf-winwlx-wlxislogoffok  
[Accessed 03 12 2022].

Microsoft, 2022. *CreateRemoteThread function (processthreadsapi.h).* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-createremotethread  
[Accessed 18 12 2022].

Microsoft, 2022. *fileapi.h header.* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/fileapi/  
[Accessed 04 12 2022].

Microsoft, 2022. *FindResourceA function (winbase.h).* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-findresourcea  
[Accessed 18 12 2022].

Microsoft, 2022. *GetAsyncKeyState function (winuser.h).* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-getasynckeystate  
[Accessed 04 12 2022].

Microsoft, 2022. *heapapi.h header.* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/heapapi/  
[Accessed 04 12 2022].

Microsoft, 2022. *memoryapi.h header.* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/memoryapi/  
[Accessed 04 12 2022].

Microsoft, 2022. *memoryapi.h header.* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/memoryapi/  
[Accessed 04 12 2022].

Microsoft, 2022. *SetWindowsHookExA function (winuser.h).* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-setwindowshookexa  
[Accessed 08 12 2022].

Microsoft, 2022. *winbase.h header.* [Online]   
Available at: https://learn.microsoft.com/en-us/windows/win32/api/winbase/  
[Accessed 04 12 2022].

Sikorski, M. & Honig, A., 2012. *Practical malware analysis : the hands-on guide to dissecting malicious software.* San Francisco: No Starch Press.