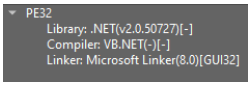
Revenge RAT Challenge

Question 1:
What compiler is used for this sample?

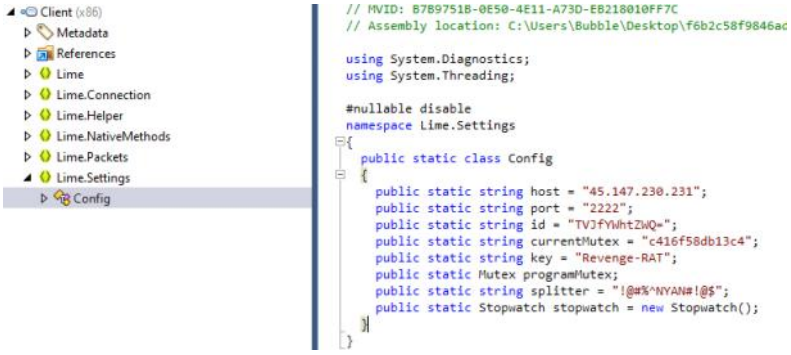
I used Detect It Easy on the malware



Answer: VB.NET

Question 2:
What is the mutex name checked by the malware at the start of execution?

I opened the malware with JetBrains dotPeek and found the Mutex under the Lime.Settings - Config



Answer: c416f58db13c4

Question 3:
What function was used to get information about the CPU?

I found the function under Lime.Helper - IdGenerator

```
public static string GetCpu()
{
  try
  {
    return Registry.GetValue("HKEY_LOCAL_MACHINE\\HARDWARE\\DESCRIPTION\\SYSTEM\\CENTRALPROCESSOR\\0", "ProcessorNameString", (object) null).ToString();
  }
  catch
  {
    return "N/A";
  }
}
```

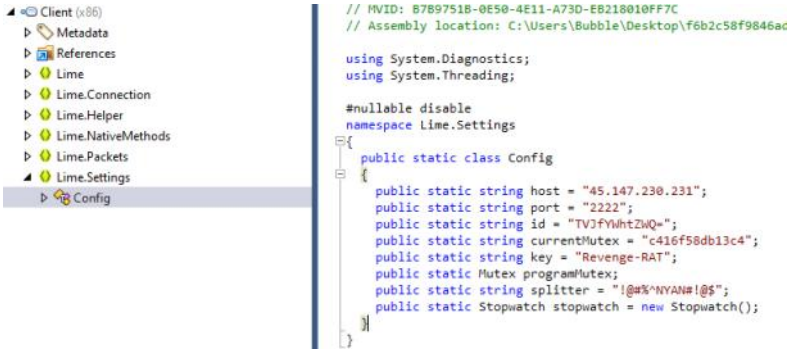
Answer: GetCpu

Question 4:
What key was used during the “SendInfo” function?

In the same code from question 3 I saw the SendInfo() function but I didn’t find any answer for this
so I used the hint

```
public static string SendInfo()
{
  return "Information" + Config.key + Config.id + Config.key + StringConverter.Encode("_" + IdGenerator.GetHardDiskSerialNumber()) + Config.key + IdGenerator.GetIp() +
}
```

The hint said to check the malware configuration which this was same like question 2



The answer is for this is "revenge-rat" I don't have a clue why and how.

Answer: revenge-rat

Question 5:
What API was used by the malware to prevent the system from going to sleep?

I found this under the PreventSleep - Run()

```
using Lime.NativeMethods;

#nullable disable
namespace Lime.Helper
{
  public static class PreventSleep
  {
    public static void Run()
    {
      try
      {
        int num = (int) Native.SetThreadExecutionState(PreventSleep.EXECUTION_STATE.ES_CONTINUOUS | PreventSleep.EXECUTION_STATE.ES_DISPLAY_REQUIRED | PreventSleep.EXECUTION_STATE.ES_SYSTEM_REQUIRED);
      }
      catch
      {
      }
    }

    public enum EXECUTION_STATE : uint
    {
      ES_SYSTEM_REQUIRED = 1,
      ES_DISPLAY_REQUIRED = 2,
      ES_CONTINUOUS = 2147483648, // 0x80000000
    }
  }
}
```

Answer: SetThreadExecutionState

Question 6:
What variable stores the volume name and the function that imported the "GetVolumeInformationA" api?

I use the searched on GetVolumeInformationA and then found only one result with this function

```
using Lime.Helper;
using System;
using System.Runtime.InteropServices;
using System.Text;

#nullable disable
namespace Lime.NativeMethods
{
  public static class Native
  {
    [DllImport("kernel32", EntryPoint = "GetVolumeInformationA", CharSet = CharSet.Ansi, SetLastError = true)]
    public static extern int GVI(
      [MarshalAs(UnmanagedType.VBByRefStr)] ref string IP,
      [MarshalAs(UnmanagedType.VBByRefStr)] ref string V,
      int T,
      ref int H,
      ref int Q,
      ref int G,
      [MarshalAs(UnmanagedType.VBByRefStr)] ref string J,
      int X);

    [DllImport("user32", EntryPoint = "GetForegroundWindow", CharSet = CharSet.Ansi, SetLastError = true)]
    public static extern IntPtr GFW();

    [DllImport("user32", CharSet = CharSet.Auto, SetLastError = true)]
    public static extern int GetWindowText(IntPtr hWnd, StringBuilder lpString, int cch);

    [DllImport("avicap32.dll", CharSet = CharSet.Ansi, SetLastError = true)]
    public static extern bool capGetDriverDescriptionA(
      short wDriver,
      [MarshalAs(UnmanagedType.VBByRefStr)] ref string lpszName,
      int cbName,
      [MarshalAs(UnmanagedType.VBByRefStr)] ref string lpszVer,
      int cbVer);

    [DllImport("kernel32.dll", SetLastError = true)]
    public static extern PreventSleep.EXECUTION_STATE SetThreadExecutionState(
      PreventSleep.EXECUTION_STATE esFlags);
  }
}
```

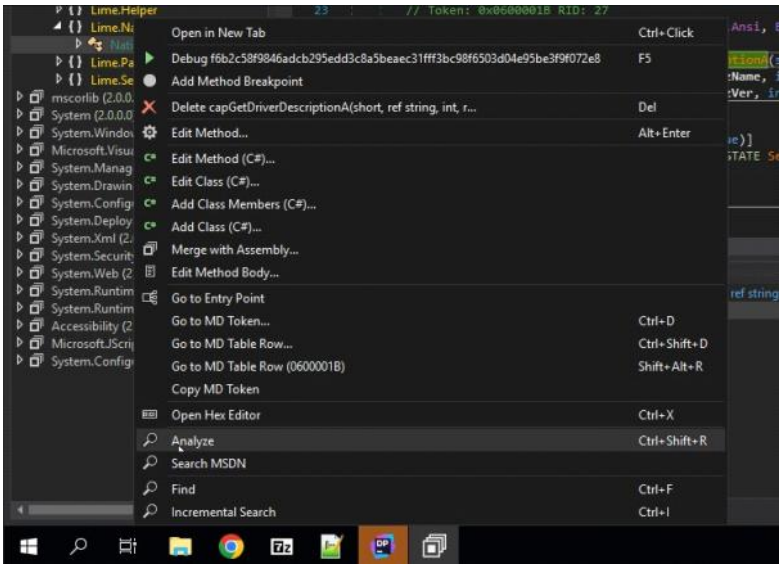
I asked the ChatGPT

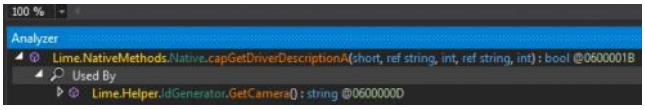
Answer: IP

Question 7:
What function was used to retrieve information about installed video capture drivers?

This question was a little tricky for me so after I used couple of methods I used another tool dnSpy and also searched for the catpGetDriverDescriptionA from the Hint

Then I right clicked on it and clicked on Analyze

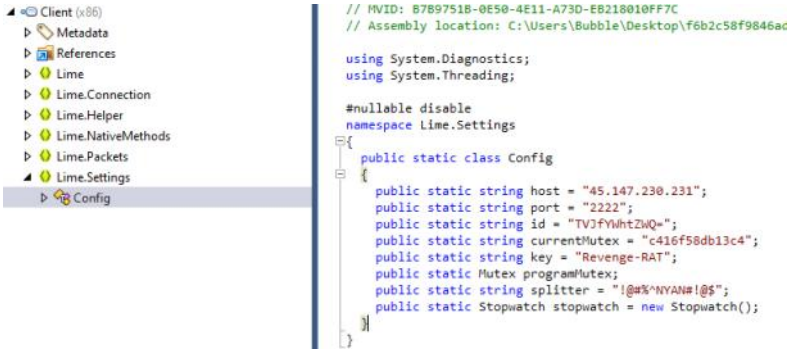


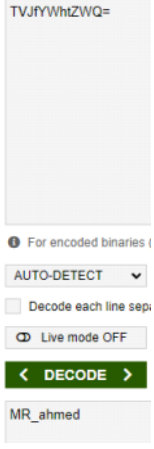


Answer: GetCamera

Question 8:
What is the value of the ID after removing obfuscation?

From task 1 I also found a small Base64 and decoded it





Answer: MR_ahmed