# LimeRAT Malware Analysis: Extracting the Config

any.run/cybersecurity-blog/limerat-malware-analysis/

hardee

# HomeMalware Analysis

LimeRAT Malware Analysis: Extracting the Config In today's article, we're going to look under the hood of a modular RAT — LimeRAT. Let's get right into it!

### What is LimeRat

LimeRAT is a Remote Access Trojan (RAT) that's been around for a few years now. It's a versatile piece of malware designed to give attackers control over an infected system. With its relatively small file size, it tries to fly under the radar of traditional antivirus solutions.

LimeRAT malware

What makes LimeRAT particularly interesting is its ability to perform a wide range of malicious activities. Some of these include keylogging, stealing passwords, and capturing screenshots. Additionally, LimeRAT can execute arbitrary commands, download and upload files, and even use the infected machine for crypto-mining or DDoS attacks.

#### LimeRAT malware analysis

To start, let's open a sample in Detect It Easy:

Detect It Easy v3.05 [Windows 10 Version	on 2009](x86_64)	—		
File name C:\sample.bin				
File type   Entry point     PE32   00408cbe	Base address           Disasm         00400000	Memory map	File info	
PE         Export         Import           Sections         Time date stamp           0003         >         2023-02-08 00:04:45	Resources     .NET     TLS       Size of image     Resources       0003a000     Manifest	Overlay Version	Hash Strings Entropy	Figure 1: sample
Scan Automatic	Endianness Mode Architecture LE 32-bit I386	Type GUI	Hex Signatures	
<ul> <li>PE32</li> <li>Library: .NET(v4.0.30319)[-]</li> <li>Compiler: VB.NET(-)[-]</li> <li>Linker: Microsoft Linker(11.0)[GUI3</li> </ul>	2]	S ? S ? S ?	Demangle Shortcuts Options	
Signatures       Directory       100%	Deep scan Recursive scan All types	Scan	About Exit	

#### overview in DiE

Upon inspection, we observe that the code has been obfuscated (MITRE T1027) and unreadable: the names of classes, methods, and variables are made out of random glyphs.

Since the sample is written in a .NET language, let's open it in DnSpy.



Figure 2: sample overview in DnSpy; note that use of obfuscation techniques

# Finding the configuration

After examining the malware's classes, we find something resembling a class with its configuration:

<pre>manespace 個大成為的法 { // Token: 0x0000000C RID: 12 public class 解学生场人顧知王公公孫子的 { // Token: 0x00000008 RID: 8 public static string 澤迎總理孫地域指司成生若為 = "At2C9Qk3d7SA7+3KqcaDzAGk3UjkKgbD1CC2tXzgMmvXISV8gQCyC4DHdLLTVSy/"; // Token: 0x04000008 RID: 1 public static string 人太玉大管司; // Token: 0x04000008 RID: 10 public static string 内斯建築商前的首司命; // Token: 0x04000008 RID: 11 public static string 内斯基案的構成主義 = "1'N']"; // Token: 0x04000000 RID: 12 public static string 行動業務金成使主義 = "['N']"; // Token: 0x04000000 RID: 13 public static string 人活顕軟指命成使大菌 - "['N']"; // Token: 0x04000000 RID: 13 public static string 人活顕軟指倫 = "['U']"; // Token: 0x04000000 RID: 15 public static string 人活顕軟指命的的司法受 = "checker netflix.exe"; // Token: 0x04000000 RID: 16 public static string 人活顕軟指合派合於全受; // Token: 0x04000000 RID: 16 public static of 開起: 16 public static boli 層後的的程度感動的法大商接 = Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 17 public static boli 層線性等行為程程的有描的目示 = Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 18 public static boli 層線性等行為程程的資源。 // Token: 0x04000002 RID: 11 public static boli 層線性等行為程程的資源。 Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 17 public static boli 層線性等行為程序的目標。 Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 17 public static boli 滑減總的目示 E conversions.ToBoolean("True"); // Token: 0x04000001 RID: 17 public static boli 滑減總的目示。 Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 17 public static boli 產物的程度有自常的目標 = Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 17 public static boli 產物的程度有自常的目標。 Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 17 public static boli 產物的程度資源的目標。 Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 17 public static boli 產物的程度有自常的目標。 Conversions.ToBoolean("True"); // Token: 0x04000001 RID: 19 public static boli 合称的程度有自常的目標。 Conversions.ToBoolean("True"); // Token: 0x0400001 RID: 17 public static boli 合称的程行有自常的目標。 Conversions.ToBoolean("True"); // Token: 0x0400001 RID: 19 public static boli 合称的程度有自常的目標。 Conversions.ToBoolean("True"); //</pre>	顧受生孫	人顧商玉公公孫子的 ×
<pre>// Token: 6x0400000C RID: 12 public class 嚴空生孫人願商玉公公孫子的 // Token: 0x04000008 RID: 8 public static string 澤地蘭程孫地說君成生希為 = "At2C9Qk3d7SA7+3Kqca0zAGk3UjkKgb01CC2tXzgWnxXISV8gQCyC40HdLLTVSy/"; // Token: 0x04000008 RID: 9 public static string 深地蘭程孫地說君成生希為 = "At2C9Qk3d7SA7+3Kqca0zAGk3UjkKgb01CC2tXzgWnxXISV8gQCyC40HdLLTVSy/"; // Token: 0x04000008 RID: 1 public static string 人大王太宮司; // Token: 0x04000008 RID: 11 public static string 内心重称生人 = "20.199.13.167"; // Token: 0x04000000 RID: 12 public static string 行顧家將命成的太媚 = "['N']"; // Token: 0x04000000 RID: 13 public static string 人大重敏或律師的司法受 = "Checker netflix.exe"; // Token: 0x04000000 RID: 15 public static string 人子攝敵成的的司法受 = "Checker netflix.exe"; // Token: 0x04000000 RID: 15 public static Mutex 号空城司首公公金受; // Token: 0x04000001 RID: 16 public static bool 薄益的的理话的者所合麵 = Conversions.ToBoolean("True"); // Token: 0x04000011 RID: 17 public static bool 二 10 public static bool 二 10 public static bool 二 11 public static bool 二 12 public static bool 二 13 public static bool 二 14 public static bool 二 15 public static bool 二 15 public static bool 二 16 public static bool 二 17 public static bool 二 18 public static bool 二 19 public static bool 二 19 public static bool □ RID: 14 public static bool □ RID: 15 public static bool □ RID: 16 public static bool □ RID: 17 public static bool □ RID: 18 // Token: 0x04000011 RID: 17 public static bool □ RID: 18 // Token: 0x04000011 RID: 17 public static bool □ RID: 18 // Token: 0x04000011 RID: 17 public static bool □ RID: 18 // Token: 0x04000011 RID: 17 public static bool □ RID: 18 // Token: 0x04000011 RID: 17 public static bool □ RID: 18 // Token: 0x04000011 RID: 17 public static bool □ RID: 18 // Token: 0x04000011 RID: 18 // Token: 0x04000011 RID: 19 // Token</pre>		namespace 顧太成為成法
<pre>// Token: 8x02000000 RID: 12 public class 酸全生孫人顧吻玉公公孫子的 { // Token: 8x04000000 RID: 8 public static string 澤地藥程孫也說言成生希為 = "At2C9Qk3d7SA7+3KqcaDzAGk3UjkKgbD1CC2tXzgWnxXISV8gQCyC4DHdLLTVSy/"; // Token: 8x04000009 RID: 9 public static string 八大玉太曾司; // Token: 8x04000008 RID: 10 public static string 的敏人墨弥生人 = "20.199.13.167"; // Token: 8x04000000 RID: 11 public static string 衍敏《墨弥生人 = "20.199.13.167"; // Token: 8x04000000 RID: 12 public static string 衍敏《墨弥生人 = "20.199.13.167"; // Token: 8x04000000 RID: 13 public static string 衍敏《墨弥生人 = "1'N' "; // Token: 8x04000000 RID: 14 public static string 行敏游略命成世太福 = " 'N' "; // Token: 8x0400000 RID: 13 public static string 八左續於前齡的的司法受 = "checker netflix.exe"; // Token: 8x0400000F RID: 15 public static string 人子導敵說的的司法受 = "checker netflix.exe"; // Token: 8x0400000F RID: 16 public static string 人子導敵說的的司法受 = "checker netflix.exe"; // Token: 8x0400000F RID: 15 public static string 人子導敵說的的司法受 = "checker netflix.exe"; // Token: 8x0400000F RID: 16 public static string 人名爾爾吉公公会受; // Token: 8x0400000F RID: 17 public static string 人名爾爾吉公公会受; // Token: 8x0400000F RID: 18 public static string 人名爾克吉公公会受; // Token: 8x0400000F RID: 17 public static bool 澤金的形理握切自為孫行合藥 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 18 public static bool 泽金的影響音和音子奇響的引將 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 18 public static bool 潘公的影響音和音子奇響的引將 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 18 public static bool 潘公能望行音音声的引將 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 18 public static bool 潘公能望行音音声的引將 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 18 public static bool 承益登書前部引將 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 18 public static bool 汤流街行音章 尔母音声称引將 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 19 public static bool 奇術說是得意意的引將 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 19 public static bool 奇術說出行音章 深日的引將 = Conversions.ToBoolean("True"); // Token: 8x04000017 RID: 19 public static bool 奇</pre>		
<pre>public class 嚴全任為人顧向王公公孫子的 { // Token: 0x04000008 RID: 8 public static string 澤地酸超落地或接司成生希為 = "At2C9Qk3d7SA7+3KqcaDzAGk3UjkKgbD1CC2tXzgMmxXISV8gQCyC4DHdLLTVSy/"; // Token: 0x04000009 RID: 9 public static string 人无长官司; // Token: 0x04000000 RID: 10 public static int 為首王的接取向前的自司命; // Token: 0x04000000 RID: 11 public static string 的人攝孫生人 = "20.199.13.167"; // Token: 0x04000000 RID: 12 public static string 行動影響命成使大師 = " 'N' "; // Token: 0x04000000 RID: 13 public static string 行動影響命成使大師 = " 'N' "; // Token: 0x04000000 RID: 13 public static string 人法嚴執自接 = " 'N' "; // Token: 0x04000000 RID: 14 public static string 人法嚴執自接 = "'\'N' "; // Token: 0x04000000 RID: 15 public static string 人法嚴執自接 = "'\'N' "; // Token: 0x04000000 RID: 14 public static string 人法嚴執自接 = "'\'N' "; // Token: 0x04000000 RID: 15 public static string 人法嚴執自接 = "\'N' "; // Token: 0x04000000 RID: 15 public static string 人法嚴執自接 = "\'N' "; // Token: 0x04000000 RID: 15 public static string 人法嚴執自接 = "\'N' "; // Token: 0x04000000 RID: 15 public static string 人法嚴執自接 = "\'N' "; // Token: 0x04000000 RID: 15 public static string 人法嚴執自接 = "\'N' "; // Token: 0x04000000 RID: 15 public static string 人法嚴執自接 = "\'N' "; // Token: 0x04000000 RID: 15 public static string 人法嚴執自接 = "\'N' "; // Token: 0x04000000 RID: 15 public static box ?@gwandsch @ = Conversions.ToBoolean("True"); // Token: 0x04000000 RID: 18 public static bool 深台的程序的目標 = Conversions.ToBoolean("True"); // Token: 0x040000012 RID: 18 public static bool @k00012 RID: 18 public static bool @k00012 RID: 18 public static bool @k0000012 RID: 18 public static bool @k000012 RID: 18 public static bool @k000012 RID: 18 public static bool @k000012 RID: 19 public static bool @k0000012 RID: 19 public static bool @k000012 RID: 19 public static bool @k000012 RID: 19 public static bool @k0000012 RID: 19 public static bool @k0000012 RID: 19 public static bool @k0000013 RID: 19</pre>		// Token: 0x0200000C RID: 12
<pre>10 { 11 // Token: 0x04000008 RID: 8 12 public static string 澤地釀EKBuckaBaukaBai = "At2C9Qk3d7SA7+3KqcaDzAGk3UjkKgbD1CC2tXzgWnvXISV8gQCyC4DHdLLTVSy/"; 14 // Token: 0x04000009 RID: 9 15 public static string 人法正法言词; 17 // Token: 0x04000006 RID: 10 18 public static string DWALSWERD = "20.199.13.167"; 19 // Token: 0x04000000 RID: 12 19 // Token: 0x04000000 RID: 13 17 // Token: 0x04000000 RID: 13 17 // Token: 0x04000000 RID: 14 18 // Token: 0x04000000 RID: 13 19 // Token: 0x04000000 RID: 14 19 // Token: 0x04000000 RID: 14 19 // Token: 0x04000000 RID: 14 10 // Token: 0x04000000 RID: 14 10 // Token: 0x04000000 RID: 14 10 // Token: 0x04000000 RID: 15 11 // Token: 0x04000000 RID: 16 12 // Token: 0x04000000 RID: 16 13 // Token: 0x04000000 RID: 16 14 // Token: 0x04000000 RID: 16 15 // Token: 0x04000000 RID: 16 16 // Token: 0x04000000 RID: 16 17 // Token: 0x04000000 RID: 16 18 // Token: 0x04000000 RID: 16 19 // Token: 0x04000000 RID: 16 10 // Token: 0x04000000 RID: 16 10 // Token: 0x04000000 RID: 16 11 // Token: 0x04000000 RID: 16 12 // Token: 0x04000000 RID: 16 13 // Token: 0x04000000 RID: 16 14 // Token: 0x04000000 RID: 16 15 // Token: 0x04000000 RID: 16 16 // Token: 0x04000000 RID: 16 17 // Token: 0x04000000 RID: 16 18 // Token: 0x04000000 RID: 16 19 // Token: 0x04000000 RID: 16 19 // Token: 0x04000000 RID: 16 10 // Token: 0x04000000 RID: 16 10 // Token: 0x04000000 RID: 17 10 // Token: 0x04000000 RID: 18 10 // Token: 0x04000000 RID: 16 10 // Token: 0x04000000 RID: 16 10 // Token: 0x04000000 RID: 17 10 // Token: 0x04000000 RID: 18 10 // Token: 0x040000000 RID: 18 10 // Token: 0x04000000 RID: 18 10 // Token: 0x04000000 RID: 18 10 // Token: 0x04000000 RID: 18 10 // Token: 0x040000000 RID: 18 10 // Token: 0x040000000 RID: 18 10 // Token: 0x04000000000000 RID: 18 10 // Token: 0x040000000000000000000000000000000000</pre>		public class 顧受生孫人顧商玉公公孫子的
11       // Token: 0x04000000 RID: 0         12       public static string 淡地離程孫地成接司成生希為 = "At2C9Qk3d7SA7+3KqcaDzAGk3UjkKgbD1CC2tXzgMmvXISV8gQCyC4DHdLLTVSy/";         13       // Token: 0x04000000 RID: 9         14       // Token: 0x04000000 RID: 10         15       public static int 為首五的接取商商的首司命;         16       // Token: 0x04000000 RID: 11         17       // Token: 0x04000000 RID: 12         18       public static string 的吸人繼孫生人 = "20.199.13.167";         14       public static string 衍動家鄉命成使太童 = "['N']";         15       // Token: 0x04000000 RID: 12         16       public static string 行顧家將命成使太童 = "['N']";         17       // Token: 0x04000000 RID: 13         18       public static string /黃癫痫的的词法受 = "['N']";         17       public static string /黃癫痫的的问意法受 = "checker netflix.exe";         17       // Token: 0x04000000 RID: 13         18       public static wtex 引受城面的公司法受 = "checker netflix.exe";         19       public static wtex 引受城面的公会会会;         17       // Token: 0x04000000 RID: 15         18       public static wtex 引受城面台公会会会;         17       // Token: 0x04000001 RID: 15         18       public static bool 牽金的的程導戰動的法太南接 = Conversions.ToBoolean("True");         19       public static bool 歐強能對了為這是的意情會當的目標。 Conversions.ToBoolean("True");<		{
<pre>public static string , MacQuargegeDack=Dack=Dack=Dack=Dack=Dack=Dack=Dack=</pre>	11	// IOKEN: 0X04000008 RID: 8
14       // Token: 0x04000009 RID: 9         15       public static string 人太玉太管司;         17       // Token: 0x0400000A RID: 10         18       public static string 的能人重原生人 = "20.199.13.167";         20       // Token: 0x0400000E RID: 11         11       public static string 的能人重原生人 = "20.199.13.167";         23       // Token: 0x0400000E RID: 12         24       public static string 行廠家將命成使太童 = "['N']";         25       // Token: 0x0400000E RID: 13         26       // Token: 0x0400000E RID: 13         27       public static string 人法睡敏的語話差 = "['N']";         28       // Token: 0x0400000E RID: 13         29       // Token: 0x0400000E RID: 14         29       // Token: 0x0400000E RID: 15         29       // Token: 0x0400000E RID: 15         29       public static string 人活動動的話送 = "checker netflix.exe";         31       public static boll 深敏的的意法会 = "checker netflix.exe";         32       // Token: 0x0400000E RID: 15         33       public static boll 深敏的印度海鲸的法太商接 = Conversions.ToBoolean("True");         33       // Token: 0x0400001 RID: 17         34       // Token: 0x0400001 RID: 17         35       // Token: 0x0400001 RID: 18         41       // Token: 0x04000012 RID: 18         42	12	public static string 達地論性分泌的成合則成土作為 = At2C90k30/3A/+3KgCaD2AGK30JKKgDUtC2CK2gWnVA13V0gUCyC4DnaLL1V3// ;
15       public static string 人太玉太管司;         17       // Token: 0x0400000A RID: 10         18       public static int 為首玉的接敬商商的首司命;         20       // Token: 0x0400000 RID: 11         21       public static string 的敬人羞孫生人 = "20.199.13.167";         23       // Token: 0x0400000 RID: 12         24       public static string 行師家海命成使太童 = " 'N' ";         25       // Token: 0x04000000 RID: 13         26       // Token: 0x04000000 RID: 13         27       public static string 人活餓執首接 = " 'L' ";         28       // Token: 0x0400000 RID: 13         29       // Token: 0x0400000 RID: 14         29       // Token: 0x0400000 RID: 15         29       // Token: 0x0400000 RID: 15         30       public static string 人子鑤爾前成的的司法受 = "checker netflix.exe";         31       // Token: 0x0400000 RID: 15         32       // Token: 0x0400000 RID: 16         33       public static bool 澤虛的的眉瞰顫的法太商接 = Conversions.ToBoolean("True");         34       // Token: 0x0400001 RID: 17         35       // Token: 0x0400001 RID: 17         36       public static bool 爾敏也道為自起意的意識。 Conversions.ToBoolean("True");         34       // Token: 0x04000011 RID: 18         34       // Token: 0x04000012 RID: 18         35 <td< td=""><td>14</td><td>// Token: 0x04000009 RID: 9</td></td<>	14	// Token: 0x04000009 RID: 9
16       // Token: 0x0400000 RID: 10         17       // Token: 0x0400000 RID: 11         18       public static int 為自玉的接敬向向的自司命;         19       // Token: 0x0400000 RID: 11         11       public static string 的敬人繼孫生人 = "20.199.13.167";         12       // Token: 0x0400000 RID: 12         14       public static string 行職家將命成使太顧 = " 'N' ";         15       // Token: 0x0400000 RID: 13         17       public static string 人法顧報首接 = " 'L' ";         18       // Token: 0x0400000 RID: 14         19       public static string 人子導顧成的的司法受 = "checker netflix.exe";         11       // Token: 0x0400000 RID: 15         12       // Token: 0x0400000 RID: 15         13       public static Mutex 引受城司官公公金受;         14       // Token: 0x0400001 RID: 15         15       public static bool 澤金的的理導戰顧的法太商接 = Conversions.ToBoolean("True");         16       public static bool 澤金的的理導戰國的法太商接 = Conversions.ToBoolean("True");         17       jublic static bool 副戰性導行指揮電的引將 = Conversions.ToBoolean("True");         18       // Token: 0x04000012 RID: 18         19       public static bool 公战的國戰聖望行導首牽的引將 = Conversions.ToBoolean("True");         14       // Token: 0x04000012 RID: 19         14       // Token: 0x04000013 RID: 19         14       /	15	public static string 人太玉太管司,
17       // Token: 0x0400000A RID: 10         18       public static int 為首玉的接敬向向的自司命;         19       // Token: 0x040000B RID: 11         11       public static string 的敬人繼孫生人 = "20.199.13.167";         12       // Token: 0x0400000C RID: 12         13       public static string 行顧家將命成使太顧 = " `N' ";         14       public static string 行顧家將命成使太顧 = " `N' ";         15       // Token: 0x0400000 RID: 13         16       // Token: 0x0400000 RID: 14         17       public static string 人才顫威的的同志受 = "checker netflix.exe";         18       // Token: 0x0400000F RID: 15         19       public static thutex 引受城司官公公会受;         19       // Token: 0x0400000F RID: 16         19       public static Mutex 引受城司官公公会受;         14       // Token: 0x0400001F RID: 16         15       public static bool 澤金的印程續較較的訪太商接 = Conversions.ToBoolean("True");         17       public static bool 歐性等行為程是的為孫行意範 = Conversions.ToBoolean("True");         16       // Token: 0x04000011 RID: 17         17       public static bool 添助進行為趕當了前首。         18       // Token: 0x04000012 RID: 18         19       public static bool 公成的繼程道行首當那的引將 = Conversions.ToBoolean("True");         14       // Token: 0x04000013 RID: 19         17       public static bool 商城範述前		
18       public static int 為首生的接敬商商的首司命;         19       // Token: 0x04000008 RID: 11         11       public static string 的敬人继孫生人 = "20.199.13.167";         22       // Token: 0x0400000C RID: 12         24       public static string 行顧家將命成使太顧 = " 'N' ";         25       // Token: 0x04000000 RID: 13         26       // Token: 0x04000000 RID: 13         27       public static string 人活願執首接 = " 'L' ";         28       // Token: 0x0400000 RID: 14         29       // Token: 0x0400000 RID: 15         21       // Token: 0x0400000 RID: 15         22       // Token: 0x0400000 RID: 15         31       public static Mutex 引受城司首公公金受;         34       // Token: 0x0400000 RID: 16         35       public static bool 澤金的的理導敬顧的法太商接 = Conversions.ToBoolean("True");         36       // Token: 0x0400001 RID: 17         37       public static bool 屬也生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         36       // Token: 0x04000012 RID: 18         37       public static bool 歐磁地當行為自筆的引將 = Conversions.ToBoolean("True");         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程留行尊首筆的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID:	17	// Token: 0x0400000A RID: 10
19       // Token: 0x04000006 RID: 11         11       public static string 的敬人繼孫生人 = "20.199.13.167";         12       // Token: 0x0400000 RID: 12         14       public static string 行顧家將命成使太顧 = " 'N' ";         15       // Token: 0x0400000 RID: 13         17       public static string 人法願執首接 = " 'L' ";         18       // Token: 0x0400000 RID: 14         19       // Token: 0x0400000 RID: 14         10       public static string 人子導顧成的的司法受 = "checker netflix.exe";         11       // Token: 0x0400000 RID: 15         12       // Token: 0x0400000 RID: 15         13       public static string 人子導顧成的的司法受 = "checker netflix.exe";         14       // Token: 0x0400000 RID: 15         15       public static Mutex 引受城司首公金受;         16       // Token: 0x0400001 RID: 16         17       public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True");         17       // Token: 0x0400001 RID: 17         18       // Token: 0x0400001 RID: 17         19       public static bool 顧敏生尊行為程星的為孫行會顧 = Conversions.ToBoolean("True");         14       // Token: 0x04000012 RID: 18         17       public static bool 公成的繼程行尊首澤的引將 = Conversions.ToBoolean("True");         14       // Token: 0x04000013 RID: 19         15       public static	18	public static int 為首圡的接破商商的首句命;
<pre>// Token: 0x0400000 HB: 12 public static string 的歌人繼孫生人 = "20.199.13.167"; // Token: 0x0400000 RID: 12 public static string 行顧家將命成使太顧 = " 'N' "; // Token: 0x0400000 RID: 13 public static string 人法顧執首接 = " 'L' "; // Token: 0x0400000 RID: 14 public static string 人子導顧成的的的司法受 = "checker netflix.exe"; // Token: 0x0400000 RID: 15 public static Mutex 引受城司官公公金受; // Token: 0x0400001 RID: 16 public static bool 澤金的的理導敏顧的法太商接 = Conversions.ToBoolean("True"); // Token: 0x0400001 RID: 17 public static bool 灤金的的理導敏顧的法太商績 = Conversions.ToBoolean("True"); // Token: 0x0400001 RID: 18 public static bool 顧敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True"); // Token: 0x0400001 RID: 18 public static bool 鼠敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True"); // Token: 0x0400001 RID: 19 public static bool 匈奴敵艦捏握行尊首澤的引將 = Conversions.ToBoolean("True"); // Token: 0x04000013 RID: 19 public static bool 匈奴敵艦惶蛇帝執引行 = Conversions.ToBoolean("True");</pre>	20	1/ Takan - 0x0100000 RTD - 11
<pre>// Token: 0x04000000 RID: 12 // Token: 0x04000000 RID: 13 // Token: 0x04000000 RID: 13 // Token: 0x04000000 RID: 13 // Token: 0x04000000 RID: 14 // Token: 0x04000000 RID: 14 // Token: 0x04000000 RID: 15 // Token: 0x0400000 RID: 16 // Token: 0x04000010 RID: 16 // Token: 0x04000011 RID: 16 // Token: 0x04000011 RID: 17 // Token: 0x04000011 RID: 17 // Token: 0x04000011 RID: 17 // Token: 0x04000011 RID: 18 // Token: 0x04000011 RID: 18 // Token: 0x04000012 RID: 18 // Token: 0x04000012 RID: 18 // Token: 0x04000012 RID: 18 // Token: 0x04000013 RID: 19 // Token: 0</pre>	20	public static string 的敬人继承生人 = "20 199 13 167"。
23       // Token: 0x0400000 RID: 12         24       public static string 行顧家將命成使太顧 = " 'N' ";         25       // Token: 0x0400000 RID: 13         27       public static string 人法顧執首接 = " 'L' ";         28       // Token: 0x0400000 RID: 14         29       // Token: 0x0400000 RID: 14         30       public static string 人子導顧成的的司法受 = "checker netflix.exe";         31	22	possie state strang upon temper to constrainty )
24       public static string 行顧家將命成使太顧 = " 'N' ";         25       // Token: 0x0400000 RID: 13         27       public static string 人法顧執首接 = " 'L' ";         28       // Token: 0x040000E RID: 14         29       // Token: 0x0400000F RID: 14         30       public static string 人子導顧成的的司法受 = "checker netflix.exe";         31       // Token: 0x040000F RID: 15         32       // Token: 0x040000F RID: 15         33       public static Mutex 引受城司官公公金受;         34       // Token: 0x04000010 RID: 16         35       // Token: 0x04000010 RID: 16         36       public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000012 RID: 18         41       // Token: 0x04000012 RID: 18         42       public static bool 癲敵性導行為程是的為孫行合顧 = Conversions.ToBoolean("True");         43       // Token: 0x04000012 RID: 18         44       // Token: 0x04000013 RID: 19         45       public static bool 向城範述增近為報引行 = Conversions.ToBoolean("True");         44       // Token: 0x04000013 RID: 19         45       public static bool 向城範述增近希報引行 = Conversions.ToBoolean("True");	23	// Token: 0x0400000C RID: 12
25       // Token: 0x04000000 RID: 13         7       public static string 人法願執首接 = "['L']";         29       // Token: 0x0400000E RID: 14         30       public static string 人子導顧成的的司法受 = "checker netflix.exe";         31       // Token: 0x040000F RID: 15         32       // Token: 0x040000F RID: 15         33       public static Mutex 引受城司官公公金受;         34       // Token: 0x04000010 RID: 16         35       // Token: 0x04000011 RID: 16         38       public static bool 澤金的的程導敏顧的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000011 RID: 17         39       public static bool 顧敏生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         41       // Token: 0x04000012 RID: 18         42       public static bool 盒做的選擇自常的目將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 商城顧地首她希執引行 = Conversions.ToBoolean("True");		public static string 行顧家將命成使太顧 = " 'N' ";
26       // Token: 0x04000000 RUD: 13         27       public static string 人法職執首接 = " 'L' ";         28       // Token: 0x0400000E RID: 14         30       public static string 人子導酶成的的向司法受 = "checker netflix.exe";         31       // Token: 0x0400000F RID: 15         32       // Token: 0x0400000F RID: 15         33       public static Mutex 引受城司官公公金受;         34       // Token: 0x04000010 RID: 16         35       // Token: 0x04000010 RID: 16         36       public static bool 澤金的的程導敬範的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000012 RID: 17         9       public static bool 顧敏生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         41       // Token: 0x04000012 RID: 18         42       public static bool @成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她若執引行 = Conversions.ToBoolean("True");	25	
27       public static string 八次顧執自接 = 1 L 1;         28       // Token: 0x0400000E RID: 14         30       public static string 人子導顧成的的司法受 = "checker netflix.exe";         31       // Token: 0x0400000F RID: 15         32       // Token: 0x0400000F RID: 15         33       public static Mutex 引受城司官公公金受;         34       // Token: 0x04000010 RID: 16         36       public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000011 RID: 17         39       public static bool 顧敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 简城顧她首她希執引行 = Conversions.ToBoolean("True");		// loken: 0x04060600 RID: 13
29       // Token: 0x0400000E RID: 14         30       public static string 人子導顧成的的的司法受 = "checker netflix.exe";         31       // Token: 0x0400000F RID: 15         32       // Token: 0x0400000F RID: 15         33       public static Mutex 引受城司官公公金受;         34       // Token: 0x04000010 RID: 16         35       // Token: 0x04000010 RID: 16         36       public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000011 RID: 17         9       public static bool 顧敏生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         41       // Token: 0x04000012 RID: 18         9       public static bool 公威的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");	27	public static string 八云順执自按 =   L   ;
30       public static string 人子導顧成的的的司法受 = "checker netflix.exe";         31       // Token: 0x0400000F RID: 15         32       // Token: 0x0400000F RID: 15         33       public static Mutex 引受城司官公公金受;         34       // Token: 0x04000010 RID: 16         35       // Token: 0x04000010 RID: 16         36       public static bool 澤金的的程導敬爾的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000011 RID: 17         39       public static bool 顧敏生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         44       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她自她希執引行 = Conversions.ToBoolean("True");	29	// Token: 0x0400000E RID: 14
<pre>31 // Token: 0x0400000F RID: 15 32 public static Mutex 引受城司官公公金受; 34 35 // Token: 0x04000010 RID: 16 36 public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True"); 37 // Token: 0x04000011 RID: 17 39 public static bool 顧敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True"); 40 41 // Token: 0x04000012 RID: 18 42 public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True"); 43 44 // Token: 0x04000013 RID: 19 45 public static bool 商城鎭她首她希執引行 = Conversions.ToBoolean("True"); 41 // Token: 0x04000013 RID: 19 45 public static bool 商城鎭她首她希執引行 = Conversions.ToBoolean("True"); 41 // Token: 0x04000013 RID: 19 45 public static bool 商城鎭她首她希執引行 = Conversions.ToBoolean("True"); 41 // Token: 0x04000013 RID: 19 45 public static bool 商城鎭她首她希執引行 = Conversions.ToBoolean("True"); 41 // Token: 0x04000013 RID: 19 45 public static bool 商城鎭她首她希執引行 = Conversions.ToBoolean("True"); </pre>	30	public static string 人子導顧成的的句法受 = "checker netflix.exe";
32       // Token: 0x0400000F RID: 15         33       public static Mutex 引受城司官公公金受;         34		
33       public static Mutex 引受城司官公公金受;         34       // Token: 0x04000010 RID: 16         35       // Token: 0x04000010 RID: 16         36       public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000011 RID: 17         39       public static bool 顧敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         40       // Token: 0x04000012 RID: 18         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 商城鎭她首她蒂執引行 = Conversions.ToBoolean("True");	32	// Token: 0x0400000F RID: 15
34       // Token: 0x04000010 RID: 16         35       public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000011 RID: 17         39       public static bool 顧敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         40       // Token: 0x04000012 RID: 18         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 商城鎭她首她蒂執引行 = Conversions.ToBoolean("True");	33	public static Mutex 引受城司官公公金受;
36       public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True");         37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000011 RID: 17         39       public static bool 顧敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         40       // Token: 0x04000012 RID: 18         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她蒂執引行 = Conversions.ToBoolean("True");		// Taken- 0401000010 RTD- 16
37       // Token: 0x04000011 RID: 17         38       // Token: 0x04000011 RID: 17         99       public static bool 顧敏生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         40       // Token: 0x04000012 RID: 18         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");		// ioken, static hool 深全的的程道砌留的注土查接 = Conversions ToBoolean("Towe")・
38       // Token: 0x04000011 RID: 17         39       public static bool 顧敏生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         40       // Token: 0x04000012 RID: 18         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");	37	
39       public static bool 顧敏生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");         40       // Token: 0x04000012 RID: 18         41       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         44       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");		// Token: 0x04000011 RID: 17
40       // Token: 0x04000012 RID: 18         41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");         41       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");		public static bool 顧敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");
41       // Token: 0x04000012 RID: 18         42       public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");         43       // Token: 0x04000013 RID: 19         45       public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");	40	
42 43 44 // Token: 0x04000013 RID: 19 45 public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");	41	// loken: 0x04060012 R1D: 18
44 // Token: 0x04000013 RID: 19 45 public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");	42	public static bool Andna處在呈行身向達的方形 = Conversions. Tobootean( True );
45 public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");	44	// Token: 0x04000013 RID: 19
010/		public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");
J1 % +	91 % -	4

Figure 3: possibly, malware configuration class

We notice that this class contains a field that appears to be a string encoded using the Base64 algorithm (MITRE T1132.001):

# // Token: 0x04000008 RID: 8 public static string 澤她顧程孫她成接司成生希為 = "At2C9Qk3d7SA7+3KqcaDzAGk3UjkKgbD1CC2tXzgWnvXISV8gQCyC4DHdLLTVSy/";<sup>Figure</sup>

4: strange class field that looks like Base64 encoded string

We attempted to decode this string using CyberChef, but were unsuccessful. It is likely that the string is not only encoded but also encrypted.

Unsuccessful attempt to decode LimeRAT string with CyberChef

Figure 5: even though it seems that this string is Base64 encoded, we can't obtain data by just decoding it Looks like the string is encoded *and* encrypted. Therefore, we will attempt to analyze this string and identify any functions or instructions that reference it. To do this, we right-click on the field and select "Analyse" from the context menu (alternatively, we can select the field and use the Ctrl + Shift + R shortcut).

In the resulting window, we are interested in where the value of this string is being read. We expand the "Read by" section and see that the string is being read in two methods:



Figure 5: two x-refs to the string

that we discovered

We briefly inspect the first method but don't see anything interesting here. It appears that this method is not specifically related to the virus configuration:

try		
{ T f	<pre>Type[] types = AppDomain.CurrentDomain.Load(顧城人顧敬人席承).GetTypes(); Tor (int i = 0; i &lt; types.Length; i++)</pre>	
	<pre>foreach (MethodInfo methodInfo in types[i].GetMethods()) </pre>	
	<pre>if (Operators.CompareString(methodInfo.Name, "CN", false) == 0) {     methodInfo.Invoke(null, new object[]     {     </pre>	
	顧受生孫人顧商玉公公孫子的.人太玉太管司, 顧受生孫人顧商玉公公孫子的.為首玉的接敬商商的首司命, 孫澤的命官.金公法商行太顧顧顧尊法官人的, 孫澤的命官.首的商公顧玉首度孫公營孫	
	廠受生孫人顧商玉公公孫子的.的敵人繼孫生人, 顧受生孫人顧商玉公公孫子的.城管將的引為, 尊司的成的望顧.尊孫引願望管她家子澤(),	
	Figur 家澤的首顧將執執生成敬.太希孫行城澤她的導的合的(顧受生孫人顧商玉公公孫子的.澤她顧程孫她成接司成生希為)	re
	<pre>}); } else if (Operators.CompareString(methodInfo.Name, "MISC", false) == 0) </pre>	
	<pre>methodInfo.Invoke(null, new object[]</pre>	
	。 尊司的成的望顧.尊孫引顧望管她家子澤(), 受她孫導的望	
	<pre>} else if (Operators.CompareString(methodInfo.Name, "CL", false) == 0) {</pre>	
	<pre>methodInfo.Invoke(null, new object[] </pre>	
	顧受生孫人顧商玉公公孫子的.商城顧她首她希執引行, 顧受生孫人顧商玉公公孫子的.人子導顧成的的的司法受, 顧受生孫人顧商玉公公孫子的.城管將的引為,	
	尊可的成的望顧.合城法導子子公顧希的為係(),	

6: the first method seems useless

Let's move on to the second method. We immediately notice some interesting code where our string is being used with the method WebClient.DownloadString, which is <u>used</u> to download a string from a remote resource.



Figure 7: the second x-ref is more interesting – looks like it uses our string in WebClient.DownloadString method Before our string is passed to WebClient.DownloadString is passed through another method that clearly transforms it into something that DownloadString can consume.

Let's take a closer look at this method and see what it does to our string.

After a quick evaluation of the method, we see that it uses instances of the RijndaelManaged and MD5CryptoServiceProvider classes.

It appears that we have found the function where our string is decrypted:



Figure 8: it seems that we found a method responsible for string decryption

# LimeRAT decryption algorithm

Let's break down how the decryption algorithm works in more detail:

- Instances of the RijndaelManaged and MD5CryptoServiceProvider classes are created. If we search for the RijndaelManaged class on MSDN, we see that it is essentially an obsolete implementation of the AES encryption algorithm (MITRE T1027). The MD5CryptoServiceProvider class, as the name implies, is used to compute an MD5 hash.
- 1. An array of 32 bytes is created and initialized with zeros. This array will be used to store the AES key.
- 1. To generate the key, the MD5 hash of **another string** from the configuration class is first computed (in our case, the string is "20[.]199.13.167").

yte[32];	
md5CryptoServiceProvider.ComputeHash(法金導望合太成.將敬離	法官官為希命繼官管的子(顧受生孫人顧商玉公公孫子的.的敬人繼孫生人));
ay, 0, array, 0, 16);	// Token: 0x0400000B RID: 11
ay, 0, array, 15, 16);	public static string 的敬人繼孫生人 = "20.199.13.167";
= array;	

Figure 9: another string from the configuration class is used to generate the AES key

- 1. Next, the first 15 bytes and then the first 16 bytes of the computed hash are copied to the previously created array. The last element of the array remains zero.
- 1. The generated key is set to the key property of the RijndaelManaged instance. The Mode property is set to CipherMode.ECB.
- 1. Finally, the original string is decoded using the Base64 algorithm and decrypted using the AES256-ECB algorithm.

Let's try to replicate this algorithm in CyberChef to confirm our findings. We will need 2 CyberChef tabs, one where we'll use MD5 to generate the AES key, and another where we'll attempt to decrypt the data.

Generating the key:

Generating the LimeRAT key
Decrypting the data:

Decrypting the data:

Recipe			Î	Input
From Base64			⊘ 11	At2C9Qk3d7SA7+3KqcaDzAGk3UjkKgbD1CC2tXzgWnvXISV8gQCyC4DHdLLTVSy/
Alphabet A-Za-z0-9+/=			•	
✓ Remove non-alpha	bet chars 🗌 St	rict mode		
AES Decrypt			⊗ II	
Key e51c4b487ef87423d	9d38b9bcb71dbe51	.c4b4	HEX 🕶	
IV			HEX 🕶	
Mode ECB	Input Raw	Output Raw		Output end: 33 length: 0
				https://pastebin.com/raw/sxNJt2ek

Figures 9, 10: we got the idea of the decryption algorithm right, as we were able to reproduce it in CyberChef After decrypting the string, we see a link to a PasteBin note: https://pastebin[.]com/raw/sxNJt2ek. When we navigate to the link, we see the C2 address of the malware.

LimeRATs C2 using data that we decrypted	
	Figure 10: we found LimeRATs C2 using data that we decrypted

# Wrapping Up

In this article, we successfully analyzed LimeRAT and uncovered its configuration. We identified the use of the .NET language and examined the malware classes, which revealed that obfuscation had been implemented. By meticulously inspecting these classes, we determined the decryption algorithm employed to decode the string containing the C2 address.

# IOCs

# Analyzed files:

SHA1	14836dd608efb4a0c552a4f370e5aafb340e2a5d
SHA256	6d08ed6acac230f41d9d6fe2a26245eeaf08c84bc7a66fddc764d82d6786d334
MD5	d36f15bef276fd447e91af6ee9e38b28
SSDEEP	3072: DDiv2GSyn 88sH 888wQ 2wm VgMk/211h 36v EclyNTY4WZd/w1UwlwEoTqPMinXHx+i: XOayyang Angel A

#### IPv4:

IOC	Description		
20[.]199.13.167:8080	LimeRAT's C2 server		

#### Domains:

IOC	Description
	Dente Die waard het Liese DAT te bide ite anieje al 00 aan van

# https://pastebin[.]com/raw/sxNJt2ek PasteBin used by LimeRAT to hide its original C2 server

### MITRE (ARMATTACK):

Tactic	Technique	Description
TA0005: Defense Evasion	T1027: Obfuscated Files or Information	Malware is using obfuscator to strip its method names, class names, etc.
TA0005: Defense Evasion	T1027: Obfuscated Files or Information	Malware uses Base64 algorithm to encode and decode data
TA0005: Defense Evasion	T1027: Obfuscated Files or Information	Malware uses AES algorithm to encrypt and decrypt data

Although effective, this manual process can be time-consuming. This is where interactive sandboxes, such as <u>ANY.RUN</u>, prove to be invaluable.

ANY.RUN offers a powerful and user-friendly platform for automating malware sample analysis. By enabling users to safely execute malware within a secure environment, ANY.RUN efficiently extracts configurations for malware like LimeRAT, ultimately saving security researchers precious time and resources.

Let us show you how our interactive sandbox can fit into your workflow — book a demo with our friendly sales team.

Interested in more content like this?

- Read our analysis of Formbook/XLoader
- Learn how we used a sandbox to analyze CryptBot
- Or check out our deep dive into Orcus Rat

Reverse Engineer. Malware Analyst at ANY.RUN

hardee

I contribute to open source from time to time and I am always up for a challenge.

malware malware analysis Juser avatar hardee Reverse Engineer, Malware Analyst at ANY.RUN I contribute to open source from time to time and I am always up for a challenge. What do you think about this post?

9 answers

- Awful
- Average
- Great

No votes so far! Be the first to rate this post.

0 comments