LimeRAT Malware Analysis: Extracting the Config

By The Hacker News

Published: 2023-04-27 · Archived: 2026-04-05 19:59:22 UTC

Remote Access Trojans (RATs) have taken the third leading position in ANY. RUN's Q1 2023 report on the most prevalent malware types, making it highly probable that your organization may face this threat.

Though LimeRAT might not be the most well-known RAT family, its versatility is what sets it apart. Capable of carrying out a broad spectrum of malicious activities, it excels not only in data exfiltration, but also in creating DDoS botnets and facilitating crypto mining. Its compact footprint allows it to elude endpoint detection systems, making it a stealthy adversary. Interestingly, LimeRAT shares similarities with njRAT, which ANY.RUN ranks as the third most popular malware family in terms of uploads during Q1 2023.

ANY.RUN researchers have recently conducted an in-depth analysis of a LimeRAT sample and successfully extracted its configuration. In this article, we'll provide a brief overview of that analysis.

Collected artifacts

SHA1	14836dd608efb4a0c552a4f370e5aafb340e2a5d
SHA256	6d08ed6acac230f41d9d6fe2a26245eeaf08c84bc7a66fddc764d82d6786d334
MD5	d36f15bef276fd447e91af6ee9e38b28
SSDEEP	3072:DDiv2GSyn88sH888wQ2wmVgMk/211h36vEcIyNTY4WZd/w1UwIwEoTqPMinXHx+i:XOayy

IPv4:

IOC	Description
20[.]199.13.167:8080	LimeRAT's Command and Control server

Domains:

IOC	Description
https://pastebin[.]com/raw/sxNJt2ek	PasteBin used by LimeRAT to hide its original Command and Control server

MITRE ATT&CK®

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

ANY.RUN is running a limited-time offer, celebrating the 7th Cyberbirthdsay

ANY.RUN is an interactive cloud malware sandbox that can extract malware configs automatically for numerous families, saving researchers hours of effort.

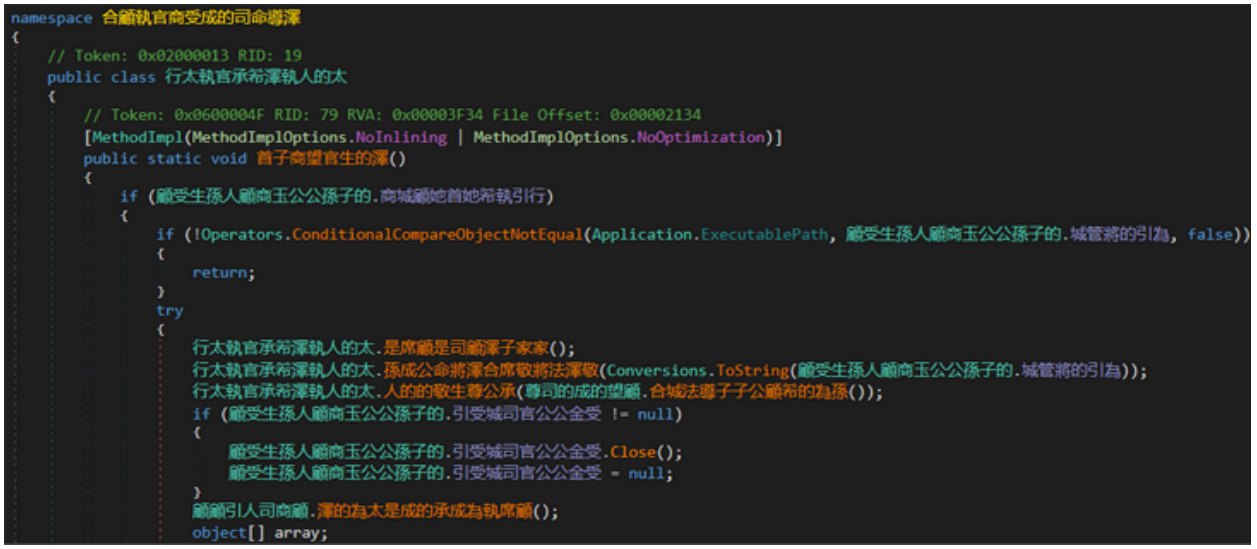
The service is celebrating its 7th anniversary and **inviting all researchers to try out advanced analysis features typically reserved for pro plans, completely free until May 5th**. This includes configuring the execution environment with Windows 8, 10, or 11.

If you discover that ANY.RUN enhances your malware analysis workflow, they are also offering a **limited promotion, available until May 5th: receive 6 or 12 months of free usage when you sign up for a yearly or two-year subscription**, respectively.

Breaking down LimeRAT's decryption algorithm

We'll share a condensed version of the article here. For a complete walkthrough and the extended analysis, head over to ANY. RUN's blog if you're interested in learning more about the workflow they employed.

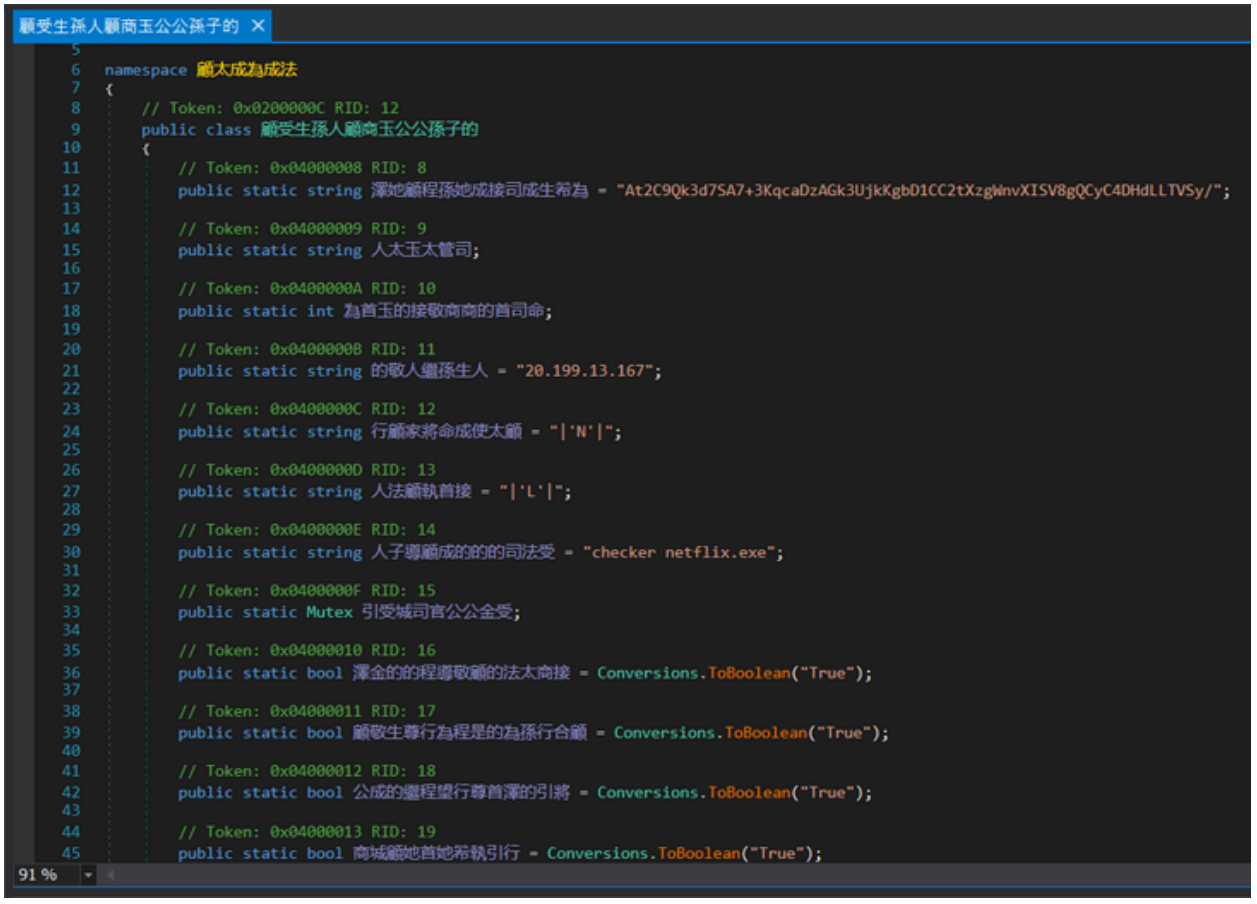
Since the sample under review was written in .NET, researchers utilized DnSpy to examine the code. Immediately, it was obvious that obfuscation techniques were being employed:



```
namespace 合顧執官商受成的司命導澤
{
    // Token: 0x02000013 RID: 19
    public class 行太執官承希澤執人的太
    {
        // Token: 0x0600004F RID: 79 RVA: 0x00003F34 File Offset: 0x00002134
        [MethodImpl(MethodImplOptions.NoInlining | MethodImplOptions.NoOptimization)]
        public static void 首子商望官生的澤()
        {
            if (顧受生孫人顧商玉公公孫子的.商城顧她首她希執引行)
            {
                if (!Operators.ConditionalCompareObjectNotEqual(Application.ExecutablePath, 顧受生孫人顧商玉公公孫子的.城管將的引為, false))
                {
                    return;
                }
                try
                {
                    行太執官承希澤執人的太.是席顧是司顧澤子家家();
                    行太執官承希澤執人的太.孫成公命將澤合席敬將法澤敬(Conversions.ToString(顧受生孫人顧商玉公公孫子的.城管將的引為));
                    行太執官承希澤執人的太.人的的敬生尊公承(尊司的成的望顧.合城法導子子公顧希的為孫());
                    if (顧受生孫人顧商玉公公孫子的.引受城司官公公金受 != null)
                    {
                        顧受生孫人顧商玉公公孫子的.引受城司官公公金受.Close();
                        顧受生孫人顧商玉公公孫子的.引受城司官公公金受 = null;
                    }
                    顧顧引人司商顧.澤的為太是成的承成為執席顧();
                    object[] array;
```

Sample overview in DnSpy; note that use of obfuscation techniques

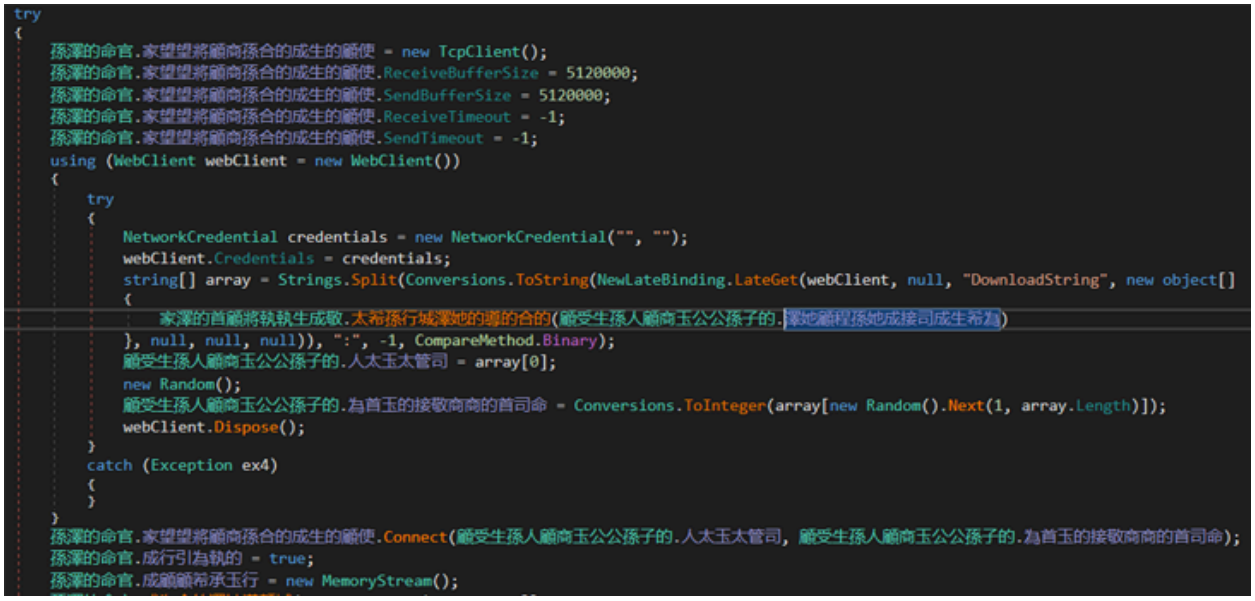
Closer examination of the code revealed a class resembling the malware configuration. Within this class, was a field containing a string that was both base64 encoded and encrypted.



```
顧受生孫人顧商玉公公孫子的

namespace 顧太成為成法
{
    // Token: 0x0200000C RID: 12
    public class 顧受生孫人顧商玉公公孫子的
    {
        // Token: 0x04000008 RID: 8
        public static string 澤她顧程孫她成接司成生希為 = "At2C9Qk3d7SA7+3KqcaDzAGk3UjkKgbD1CC2tXzgWnvXISV8gQCyC4DHdLLTVSy/";

        // Token: 0x04000009 RID: 9
        public static string 人太玉太管司;

        // Token: 0x0400000A RID: 10
        public static int 為首玉的接敬商商的首司命;

        // Token: 0x0400000B RID: 11
        public static string 的敬人繼孫生人 = "20.199.13.167";

        // Token: 0x0400000C RID: 12
        public static string 行顧家將命成使太顧 = "|'N'|";

        // Token: 0x0400000D RID: 13
        public static string 人法顧執首接 = "|'L'|";

        // Token: 0x0400000E RID: 14
        public static string 人子導顧成的的的司法受 = "checker netflix.exe";

        // Token: 0x0400000F RID: 15
        public static Mutex 引受城司官公公金受;

        // Token: 0x04000010 RID: 16
        public static bool 澤金的的程導敬顧的法太商接 = Conversions.ToBoolean("True");

        // Token: 0x04000011 RID: 17
        public static bool 顧敬生尊行為程是的為孫行合顧 = Conversions.ToBoolean("True");

        // Token: 0x04000012 RID: 18
        public static bool 公成的繼程望行尊首澤的引將 = Conversions.ToBoolean("True");

        // Token: 0x04000013 RID: 19
        public static bool 商城顧她首她希執引行 = Conversions.ToBoolean("True");
91 %
```

Possibly, malware configuration class

Continuing the code inspection, ANY.RUN researchers pinpointed a function responsible for decrypting the string. By employing the "Read by" filter in DnSpy, they tracked down methods where the string was being read, which led to a total of two methods. The first method proved unfruitful, but the second one looked interesting:

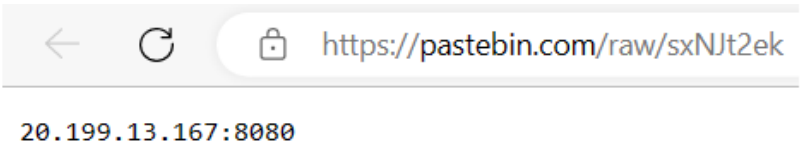


The second x-ref is more interesting. It seems that it uses our string in WebClient.DownloadString method

This method turned out to be responsible for decryption. By closely examining it, it was possible to reconstruct the process by which LimeRAT decrypts its configuration:

1. Instances of the **RijndaelManaged** and **MD5CryptoServiceProvider** classes are instantiated. As per MSDN, **RijndaelManaged** is an outdated implementation of the AES encryption algorithm (**MITRE T1027**), while **MD5CryptoServiceProvider** computes MD5 hashes.
2. A 32-byte array, initialized with zeros, is generated to store the AES key.
3. The key is created by first calculating the MD5 hash of a distinct string within the configuration class (in our analysis, the string is "20[.]199.13.167").
4. The initial 15 bytes, followed by the first 16 bytes of the calculated hash, are copied into the previously established array. The final element of the array remains zero.
5. The derived key is assigned to the key property of the **RijndaelManaged** instance, while the Mode property is configured as **CipherMode.ECB**.
6. Ultimately, the primary string undergoes decoding via the **Base64** algorithm and decryption using the **AES256-ECB** algorithm.

Decrypting the string revealed a link to a PasteBin note: **https://pastebin[.]com/raw/sxNJt2ek**. Within this note, was LimeRAT's Command and Control (C2) server:



LimeRATs C2 discovered with decrypted data

To wrap up

We hope you found this brief overview of our LimeRAT configuration decryption process insightful. For a more comprehensive examination, head over to the full article on ANY.RUN's blog, to get additional context on the steps and check the decryption process using CyberChef.

Also, remember that ANY. RUN's currently offering limited-time deals, featuring discounts on subscriptions and an expanded feature set for free plans, including the ability to configure execution environments with Windows 8, 10, and 11 operating systems. This offer expires on May 5th.

This is an ideal opportunity to test out ANY.RUN and determine if it streamlines your workflow, or to secure a subscription at an unbeatable price and reap the benefits of significant time savings through static and behavioral analysis.

To learn more about this offer, visit ANY.RUN plans.

Found this article interesting? This article is a contributed piece from one of our valued partners. Follow us on Google News, Twitter and LinkedIn to read more exclusive content we post.
